OPTIMAL DETERMINISTIC ALGORITHM FOR THE SIMPLE SYMMETRIC HYPOTHESES TESTING PROBLEM

Boorapa Chodchoey

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

OPTIMAL DETERMINISTIC ALGORITHM FOR THE
SIMPLE SYMMETRIC HYPOTHESES TESTING PROBLEM

by

Boorapa Chodchoey

September 1974

Thesis Advisor: B. O. Shubert

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Optimal Deterministic Algorithm for the Simple Symmetric Hypotheses Testing Problem		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1974
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Boorapa Chodchoey		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1974
		13. NUMBER OF PAGES 79
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automaton
Finite-Memory Deterministic Algorithm
Decision-Making
Hypothesis Testing with Finite Memory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper introduces a class of finite-memory deterministic algorithms for the following problem of hypotheses testing under a finite memory constraint. Let $X_1, X_2, X_3, \ldots$ be a sequence of independent, identically distributed Bernoulli random variables where X_i can take on values H or T. The problem is to decide between the two simple hypothesis H: $P(X_i=H)=p$ vs. T: $P(X_i=H)=q$, where $P(H \text{ is true}) = \pi_0 = P(T \text{ is true}) = \pi_1 = \frac{1}{2}$. The X_i's are

Block #20 continued

observed sequentially and a new decision must be formulated after each observation. Let the data be summarized after each new observation by a 2n-valued statistic

$$V \in S = \{1_h, 2_h, \ldots, n_h, n_t, (n-1)_t, \ldots, 2_t, 1_t\},$$

which is updated according to the rule $V_k = f(V_{k-1}, X_k)$, where $f : S \times (H,T) \rightarrow S$; is the transition function. Let the decision rule take action

$$d(V_k) = \begin{cases} H \text{ is true if } V_k \in (1_h, 2_h, \ldots, n_h) \\ T \text{ is true if } V_k \in (n_t, n-1_t, \ldots, 2_t, 1_t), \end{cases}$$

at time k. The objective is to find the function f which minimizes the probability of error

$$P(e) = \pi_o \; P(d = T|H \text{ is true}) + \pi_1 \; P(d = H|T \text{ is true}).$$

The algorithm may be taught of as a finite state automaton, in which the inputs are the observations, the outputs are the decisions, and the states constitute the memory. In this paper, the optimal algorithms are found for a small number of states (up to 20).

Optimal Deterministic Algorithm for the

Simple Symmetric Hypotheses Testing Problem

by

Boorapa Chodchoey
2nd. Lieutenant, Royal Thai Air Force
B.(Ed), The Colledge of Education, Prasanmitr, Thailand, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1974

ABSTRACT

This paper introduces a class of finite-memory deterministic algorithms for the following problem of hypotheses testing under a finite memory constraint. Let $X_1, X_2, X_3, \ldots$ be a sequence of independent, identically distributed Bernoulli random variables where X_i can take on values H or T. The problem is to decide between the two simple hypothesis $H : P(X_i = H) = p$ vs. $T : P(X_i = H) = q$, where $P(H \text{ is true}) = \pi_0 = P(T \text{ is true}) = \pi_1 = \frac{1}{2}$. The X_i's are observed sequentially and a new decision must be formulated after each observation. Let the data be summarized after each new observation by a 2n-valued statistic $V \varepsilon S = \{1_h, 2_h, \ldots, n_h, n_t, (n-1)_t, \ldots, 2_t, 1_t\}$, which is updated according to the rule $V_k = f(V_{k-1}, X_k)$, where $f : S \times (H,T) \to S$; is the transition function. Let the decision rule take action

$$d(V_k) = \begin{cases} H \text{ is true if } V_k \varepsilon (1_h, 2_h, \ldots, n_h) \\ T \text{ is true if } V_k \varepsilon (n_t, n-1_t, \ldots, 2_t, 1_t), \end{cases}$$

at time k. The objective is to find the function f which minimizes the probability of error $P(e) = \pi_0 \, P(d = T | H \text{ is true}) + \pi_1 \, P(d = H | T \text{ is true})$.

The algorithm may be taught of as a finite state automaton, in which the inputs are the observations, the outputs are the decisions, and the states constitute the memory. In this paper, the optimal algorithms are found for a small number of states (up to 20).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

The invaluable advice and assistance of Professor Bruno Shubert is gratefully acknowledged.

I. INTRODUCTION

In the near future, computers will play even more important roles in every day life than ever. We are expecting more compact machines with more working capabilities and requiring less human intervention. The idea of designing a machine with decision making capability is of great interest. If such a machine were to be employed in a decision process, then we face the problem of trading off the error caused by the machine operation against its complexity. In a smaller size machine with limited core memory, the error caused by the decision under finite memory is even more significant. The attempt of designing a small machine with limited core size which operates with least probability of error is worth of time and effort, especially if such a device is to be used for some special tasks. Consider an exploration of a distant star with an unmaned spacecraft. If a computer with a decision making capability were to be used within the spacecraft, it would almost certainly have to be small and of limited core size. This kind of machine or automaton would be required to make decision with minimum probability of error constrained by the available memory. In this example, the machine or automaton acts like an input-output machine which has the observed data as the input. The output of the machine are the decisions. Only a finite amount of informations can be stored at any time. To

construct such an automaton, a form of a decision process could probably be adapted from the statistical test of hypotheses.

Let $X_1, X_2, \ldots$ be a sequence of independent, identically distributed random observations drawn according to a probability measure T defined on arbitrary probability space. Consider the simple hypotheses testing problem

$$H_0: T = T_o \qquad \text{vs.} \qquad H_1: T = T_1 \ .$$

Let the prior probabilities of the null and alternative hypotheses be denoted by π_o and π_1 respectively. The goal is to find a sequence of decision rules $d_1(X_1), d_2(X_1, X_2), \ldots$ which minimize the asymptotic probability of error

$$P(e) = E\left[\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} e_i\right]$$

where

$$e_i = \begin{cases} 1 \text{ if } d_i \neq H \text{ true,} \\ 0 \text{ if } d_i = H \text{ true.} \end{cases} \tag{1}$$

Denoting, for a sample of size n, α_n the probability of type I error and β_n the probability of type II error it is well known that α_n and β_n will exponentially approach zero as n becomes large. However, the decision d_n depends on $(X_1, X_2, \ldots, X_n)$, so that, as n increases, the amount of data to be stored increases without bounds. Some means of data reduction may therefore be desirable. Sufficient statistics can sometimes be used to reduce the required size of memory. These statistics lose no information when used. Unfortunately,

the following example shows that this type of data reduction is sometimes misleading.

Let $X \sim N(\mu,1)$ with the hypotheses

$$H_0: \mu = 1 \qquad \text{vs.} \qquad H_1: \mu = -1$$

where

$$\pi_o = \pi_1 = \tfrac{1}{2}.$$

A sufficient statistic for this test is $Y_n = \sum_{i=1}^{n} X_i$, where Y_n is the value of the statistic after n observations. A simple optimal decision scheme is

$$d_n = \begin{cases} H_0 & \text{if } Y_n \geq 0 \\ H_1 & \text{if } Y_n < 0. \end{cases}$$

The updating rule for Y_n is given by

$$Y_{n+1} = Y_n + X_{n+1}.$$

Thus Y_n contains all the desired informations about $(X_1,X_2,\ldots,X_n)$, and only Y_n needs to be recorded at the time n. However, Y_n is real-valued so that a potentially infinite storage is needed for it alone. If the memory can store a real number then it can store any number of real numbers so that no saving is actually achieved. The attempt of rounding off the sufficient statistic to some finite number of digits need not be optimal. In fact, Cover [2] has shown that the error probability need not tend to zero if rounded off statistic is used. If constrained by finite memory, some other statistic must be devised in cooperation with some appropriate decision rule. Before further discussion

of this topic, it would be worthwhile to look at the literature survey concerning a decision model constrained by finite memory. The first mention of finite memory in connection with a statistical decision problem is to be found in [1]. This paper by Robbins discussed the problem of choosing one of two ways of action, each of which may lead to success or failure, in such a way as to maximize the long-run proportion of successes obtained. The choice each time is allowed to depend only on a finite number of past observation. Suppose we have two coins and we wish to maximize the number of heads thrown during a sequence of tosses. If we had prior knowledge of which coin has the larger probability of turning up a head, we should use it exclusively, irrespective of the outcomes of previous tosses. Then, with probability 1

$$\lim_{n\to\infty} \frac{\text{number of heads in first n tosses}}{n} = \max\,(p_1,p_2),$$

where

p_i = probability of head for the $i^{\underline{th}}$ coin.

If no information about p_1 and p_2 is available, then there still is some decision rule which asymptotically achieves the above limit. Robbins had come up with the following decision rule, said to be of type r if the decision as to which coin will be used for the $n^{\underline{th}}$ toss depends only on the results of tosses n-r,n-r+1,...,n-1. "Define the rule R_r as follows: start tossing with coin 1, stop if the first toss is a tail, otherwise continue tossing until the

first run of r consecutive tails occurs and then stop. This defines the first block of tosses with coin 1. Now start tossing with coin 2 and apply the same rule, obtaining the first block of tosses with coin 2. Then start again with coin 1 and apply the same rule, obtaining the second block of tosses with coin 1 and so on indefinitely, thus generating an infinite sequence of tosses consisting of alternate blocks of tosses with coin 1 and 2." With rule R_r, of type r so defined Robbins proved that with probability 1

$$\lim_{n\to\infty} \frac{\text{number of heads in first n tosses}}{n} = \frac{p_1(1-p_2)^r + p_2(1-p_1)^r}{(1-p_1)^r + (1-p_2)^r}$$

and note that

$$\lim_{n\to\infty} \frac{p_1(1-p_2)^r + p_2\ (1-p_1)^r}{(1-p_1)^r + (1-p_2)^r} = \max\ (p_1,p_2).$$

So the rule R_r is the best among the other of type r which maximize the long-run proportion of heads obtained.

Although Robbins in his paper did not consider hypotheses testing problem, the paper nevertheless stimulated the idea of using a finite memory in statistical decision. By using the same m-finite past memory as in [1], Cover [2] developed a decision rule using a 4 states memory alogrithm for the hypotheses testing problem as applied to Bernoulli trials. The details of this can be found in [2].

The first actual finite state memory algorithm was proposed in [3] and [4] by Hellman and Cover. Here the past observations were stored as a state of a machine constrained

to have only m states. Consider the two hypotheses testing problem as stated earlier, where

$$H_0: T = T_o \qquad \text{vs} \qquad H_1: T = T_i$$

and the prior probabilities P(H_0 is true) = π_o and P(H_1 is true) = π_1.

After each observation, let the data be summarized by an m-valued statistic $V\varepsilon\{1,2,\ldots,m\}$ where the updating scheme is $V_n = f(V_{n-1}, X_n)$; $f:\{1,2,\ldots,m\}XW\rightarrow\{1,2,\ldots,m\}$ where $X_i \varepsilon W$ the set of values which X_i can take on for $i = 1,2,\ldots$ and f is the transition function.

Let the decision rule be defined as

$$d_n = d(V_n);\ d : \{1,2,\ldots,m\} \rightarrow \{H_0,H_1\}.$$

The pair (f,d) then describes a finite-state automaton with inputs X_n and outputs $d_n = d(V_n)$ and state space S = {1,2, ...,m}. Here V_n is the state at time n. Under hypotheses H_0 or H_1, the sequence V_n, with some specific initial state, forms a Markov chain over the state space S. In order to minimize

$$P(e) = E\left[\lim_{n\rightarrow\infty} \frac{1}{n} \sum_{i=1}^{n} e_i\right]$$

where e_i is the error as defined in (1), we need to find the optimal pair (f,d). Hellman and Cover have established a lower bound for P(e) as follows.

Let f_0 and f_1 be the probability densities of the sample under the respective hypotheses with respect to a dominating measure. Define the likelihood ratio to be $\ell(x) = \dfrac{f_0(x)}{f_1(x)}$.

Let $\bar{\ell}$ denote the essential supremum of $\ell(x)$ and $\underline{\ell}$ the essential infimum of $\ell(x)$ where the supremum and infimum are taken over all measurable sets with positive dominating measure. Define $\gamma = \bar{\ell}/\underline{\ell}$. Then for an irreducible[1] m-state automaton, we have $P(e) \geq P^*$ where

$$P^* = \min \left\{ \frac{2\sqrt{\pi_o \pi_1 \gamma^{m-1}} - 1}{\gamma^{m-1} - 1}, \pi_o, \pi_1 \right\}.$$

It was further proved in [3] that the same bound P^* also hold for $P(e)$ in a reducible $(m+1)$-state automaton. For a special case where $\pi_o = \pi_1 = \frac{1}{2}$, the P^* for the irreducible m-state automaton becomes

$$P^* = \frac{1}{\gamma^{\frac{1}{2}(m-1)} + 1} .$$

If the m-state automaton is reducible, the bound becomes

$$P^* = \frac{1}{\gamma^{\frac{1}{2}(m-2)} + 1} .$$

The ratio $\gamma = \bar{\ell}/\underline{\ell}$ is a measure of the separation between the two hypotheses. Notice that P^* decreases exponentially with m.

However, Hellman and Cover have shown that, except for degenerate cases, no machine can actually achieve the bound P^*. But as an example in Bernoulli case, an ε-optimal class of automata (i.e., such that for any $\varepsilon > 0$ there exist an automaton with $P(e) \leq P^* + \varepsilon$) was introduced. The detailed

---

[1] We call the automaton irreducible (reducible) if the result Markov chain formed by sequence $\{V_n\}$ under both hypotheses is irreducible (reducible).

description of this class of automata is as follows: Let $X_1, X_2, \ldots$ be a sequence of independent, identically distributed Bernoulli random variables where X_i can take on value H or T, let the two hypotheses be

$$H : P(X_i = H) = p_h \qquad \text{vs.} \qquad T : P(X_i = T) = p_t.$$

where the prior probabilities are $\pi_o = \pi_1 = \frac{1}{2}$ and $q_i = 1 - p_i$ for $i = h,t$.

Without loss of generality it may assume that $p_h > p_t$, in which case, $\bar{\ell} = p_h/p_t$; $\underline{\ell} = q_h/q_t$ and $\gamma = \bar{\ell}/\underline{\ell} = p_h q_t / p_t q_h$. If the hypotheses are symmetric, that is if $p_h = 1 - p_t$, then $\gamma = (p_h/q_h)^2$; hence $P(e) \geq 1/\left(1 + (p_h/q_h)^{m-1}\right)$ for an irreducible m-state automaton, and $P(e) \geq 1/\left(1 + (p_h/q_h)^{m-2}\right)$ for a reducible m-state automaton.

Let the transition function f be defined as follows (see Figure 1):

$$f(i,X) = \begin{cases} i+1 & \text{if } X = H \\ i-1 & \text{if } X = T \end{cases}$$

for $i = 2,3,\ldots,m-1$

$$f(1,X) = \begin{cases} 2 & \text{with probability } \delta > 0 \text{ if } X = H, \\ 1 & \text{otherwise,} \end{cases}$$

$$f(m,X) = \begin{cases} m-1 & \text{with probability } k\delta > 0 \text{ if } X = T \\ m & \text{otherwise,} \end{cases}$$

where $k = 1$ for the case of symmetric hypotheses.

Figure 1.

The transitions are made to the adjacent states only when the event X = H or X = T are observed. Otherwise, the state does not change. This automaton will reach an end state only on strong evidence to support the corresponding hypotheses (i.e., to enter state m, preference on *H* is strongly and vice versa for *T*). However, leaving the end states has only a very small chance when δ is small. A decision made at the end states results in the smallest probability of error and as $\delta \to 0$, the error probability P(e) should asymptotically approach P*.

The Hellman-Cover algorithm is useful in providing sequences of decisions, but not very suitable for the case when only a single decision is required. The irreducible automaton will asymptotically approach the lower bound for P(e) after a "large enough" number of observations and there is no way to tell how large the number should be. In case when we need a more reliable decision we may have to wait for a long time. In addition, the automaton requires artificial randomization for generating the probabilities δ and $k\delta$ to transit out of the end states. Such a random generator needs additional memory to be added to the automaton and it will no longer be a finite state automaton if we need a very small δ as close to zero.

On the same problem of the Bernoulli case, viewed as two-ways Bernoulli classification problem, Shubert had introduced in his paper [8] a deterministic machine which can perform as well as optimal randomized machines only if the machine memory is increased by less than one bit. This class of algorithm use the data source itself to provide the necessary randomization. The problem of finding a truly deterministic algorithm has been considered by Anderson in his thesis [5]. He developed a special class of symmetric (2n+3)-state algorithms with two absorbing states. The algorithm can perform a decision process without randomization.

He defined the transition function f as follows (see Figure 2):

$$
\begin{array}{llll}
f(s,H) = s+1 , & f(s,T) = s-\rho(s) & \text{if } s = 1,2,\ldots,n \\
f(s,T) = s-1 , & f(s,H) = s+\rho(s) & \text{if } s = -1,-2,\ldots,-n \\
f(s,H) = 1 , & f(s,T) = -1 & \text{if } s = 0 \\
f(s,H) = s , & f(s,T) = s & \text{if } s = \pm\ (n+1)
\end{array}
$$

$$\text{where } 1 \leq \rho(s) \leq s.$$

This algorithm, however, has higher probability of error than the randomized machine. But it was shown in [5] that, if the randomization is provided for this deterministic machine, the Hellman-Cover lower bound for P(e) can be approached arbitrarily closely.

In this thesis, another class of the deterministic algorithm is introduced and investigated. The class is of ergodic type and it is the only characteristic in which it

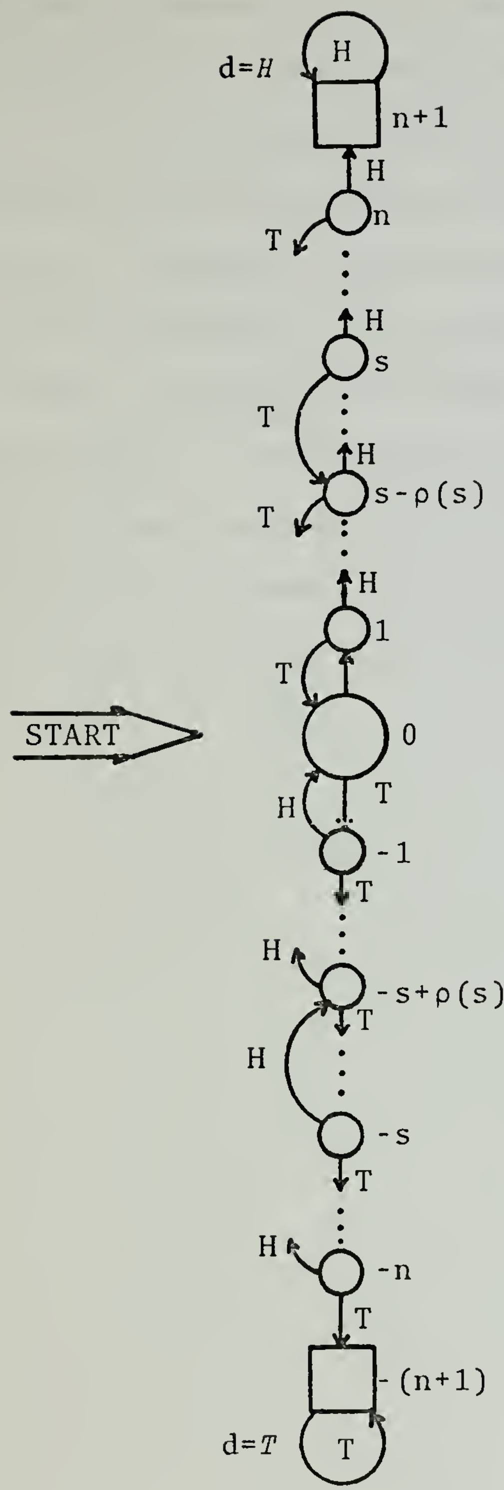


Figure 2. An Algorithm $f = \langle \rho(1), \ldots, \rho(n) \rangle$.

differs from the one discussed by Anderson [5]. Since there were a lot of similarities in this paper and Anderson's paper, the sections concerning the proof of achieving the bound with randomization and deriving the asymptotic bound for P(e) are omitted. However, the determination of the probability of error in terms of the algorithm is presented in the complete form. The search of optimal algorithms for the cases of small number of states have been done by algebraic computation in some cases and by computer search for the other. The results are summarized in Table I and Table II with Figures 5, 6 and 7 to provide clearer idea of the trend for larger number of states.

II. DESCRIPTION OF THE ALGORITHM

Let $X_1, X_2, X_3, \ldots$ be a sequence of independent identically distributed Bernoulli random variables where

$$X_i \varepsilon G = \{X : X = H \text{ or } T\} \quad \text{for all } i = 1,2,3,\ldots$$

Consider the simple hypothesis testing problem

$$\mathcal{H} : P(X_i = H) = p \quad \text{vs.} \quad \mathcal{T} : P(X_i = H) = q,$$

where

$$\tfrac{1}{2} < p < 1 \text{ and } q = 1 - p.$$

Denote the prior probability of $\mathcal{H}$ being true as π_h and denote π_t, the prior probability of $\mathcal{T}$ being true. Here only the case $\pi_h = \pi_t = \frac{1}{2}$, the symmetric hypotheses testing problem will be considered. The sequence of random variables $X_1, X_2, X_3, \ldots$ can be viewed as successive tosses of a coin which is biased towards heads under hypothesis $\mathcal{H}$ or biased towards tails under hypothesis $\mathcal{T}$.

Let $\underline{r} = \{r(2), r(3), \ldots, r(n)\}$ be a sequence of positive integers such that $1 \leq r(i) < i$ for $i = 2,3,\ldots,n$; where n is any positive integer.

With each $\underline{r}$ we associate a finite-memory symmetric algorithm (M,f,d) (See Figure 3), where M is defined to be a set of 2n states such that $M = \{(1,h),(2,h),\ldots,(n,h),(n,t),\ldots,(2,t),(1,t)\}$. The subscripts h and t indicate the states in which the decision in favor of the hypothesis $\mathcal{H}$ and $\mathcal{T}$ respectively is made. In other words, the decision rule d is $d\{(i,h)\} = \mathcal{H}$ and $d\{(i,t)\} = \mathcal{T}$ for $i = 1,2,\ldots,n$.

The transition function, f is defined by

$$\left.\begin{array}{l} f\{(i,h),H\} = (r(i),h) \\ \\ f\{(i,t),T\} = (r(i),t) \end{array}\right\} \quad \text{if } i = 2,3,\ldots,n$$

$$\left.\begin{array}{l} f\{(i,h),T\} = ((i+1),h) \\ \\ f\{(i,t),H\} = ((i+1),t) \end{array}\right\} \quad \text{if } i = 1,2,\ldots,n$$

$$f\{(1,h),H\} = (1,h)$$

$$f\{(1,t),T\} = (1,t)$$

$$f\{(n,h),T\} = (n,t)$$

$$f\{(n,t),H\} = (n,h)$$

With the sequence $X_1,X_2,X_3,\ldots$ of independent identically distributed random variables as the input, the states of the algorithm (M,f,d) form an ergodic Markov chain. The transition probabilities are

$$\left.\begin{array}{l} P\{(i,h) \to (r(i),h)\} = p \\ \\ P\{(i,t) \to (r(i),t)\} = q \end{array}\right\} \quad \text{if } i = 2,3,\ldots,n$$

$$\left.\begin{array}{l} P\{(i,h) \to (i+1,h)\} = q \\ \\ P\{(i,t) \to (i+1,t)\} = p \end{array}\right\} \quad \text{if } i = 1,2,\ldots,n-1$$

$$P\{(1,h) \to (1,h)\} = p$$

$$P\{(1,t) \to (1,t)\} = q$$

$$P\{(n,h) \to (n,t)\} = q$$

$$P\{(n,t) \to (n,h)\} = p$$

under the hypothesis H. Under the hypothesis T, the transition

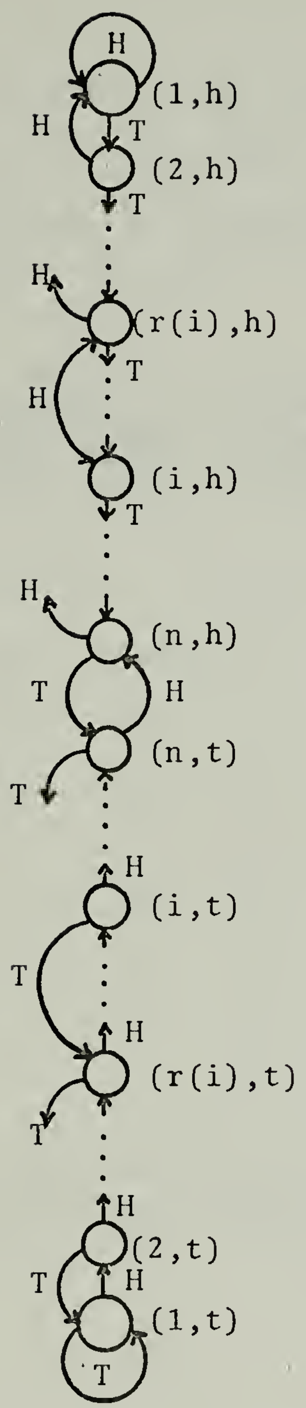
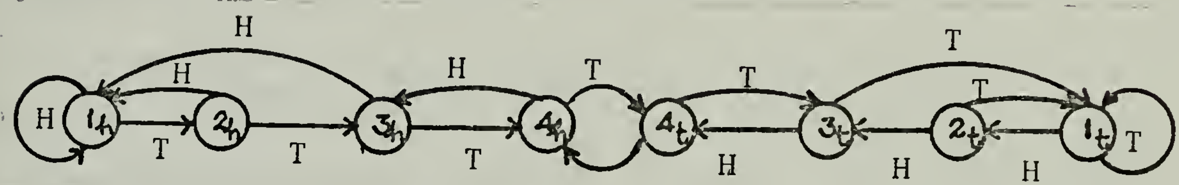


Figure 3. The Algorithm f = < r(2),r(3),...,r(n) >; where (r(i),j) is the state that the transition transits from (i,j),j = h,t.

probabilities have the same form as above with p and q are interchanged in places.

From now on, the specific form of the algorithm (M,f,d) will be denoted by $f_n = < \underline{r} > = < r(2),r(3),\ldots,r(n) >$. Figure 4 shows the algorithm and the transition matrix of the case where n = 4 and $f_4 = < 1,1,3 >$.



Transition Diagram for $f_4 = < 1,1,3 >$

	(1,h)	(2,h)	(3,h)	(4,h)	(4,t)	(3,t)	(2,t)	(1,t)
(1,h)	p	q	0	0	0	0	0	0
(2,h)	p	0	q	0	0	0	0	0
(3,h)	p	0	0	q	0	0	0	0
(4,h)	0	0	p	0	q	0	0	0
(4,t)	0	0	0	p	0	q	0	0
(3,t)	0	0	0	0	p	0	0	q
(2,t)	0	0	0	0	0	p	0	q
(1,t)	0	0	0	0	0	0	p	q

Figure 4. Illustration of Algorithm Diagram and the Transition Matrix for the Case of n = 4 and $f_4(r(2),r(3),r(4)) = < 1,1,3 >$.

III. DETERMINATION OF ERROR PROBABILITY

Let $S_h = \{(1,h),(2,h),\ldots,(n,h)\}$, let $S_t = \{(n,t),(n-1,t),\ldots,(2,t),(1,t)\}$ be the two subsets of the set of states M. Define $\mu(s)$, $s\varepsilon\, S_h \cup S_t$ be its stationary probabilities and let

$$\mu(S_h) = \sum_{s\varepsilon S_h} \mu(s) \text{ and } \mu(S_t) = \sum_{s\varepsilon S_t} \mu(s)$$

be stationary probabilities of the state being in S_h and S_t respectively. With the decision rule d, the probability of error can be written as

$$P(e) = \pi_h\, P(d = T|H) + \pi_t\, P(d = H|T)$$

$$= \tfrac{1}{2}\, P(s\varepsilon S_t|H) + \tfrac{1}{2}\, P(s\varepsilon S_h|T).$$

By the symmetry of the algorithm we have

$P(s\varepsilon S_t|H) = P(s\varepsilon S_h|T)$ that is

$$P(e) = P(s\varepsilon S_t|H)$$

$$= \frac{P(s\varepsilon S_t|H)}{P(s\varepsilon S_t|H) + P(s\varepsilon S_h|H)}$$

$$= \left\{1 + \frac{P(s\varepsilon S_h|H)}{P(s\varepsilon S_t|H)}\right\}^{-1} = \left\{1 + \frac{\mu(S_h)}{\mu(S_t)}\right\}^{-1}.$$

In order to obtain explicit expression for the probability of error in term of the algorithm f_n, we now prove the following proprosition.

<u>Proprosition 1</u>: Let $f_n = < r(2),r(3),\ldots,r(n) >$ be given. Then,

$$\frac{\mu(S_h)}{\mu(S_t)} = \left(\frac{p}{q}\right)^n \frac{A_n}{B_n} ,$$

where A_n and B_n are polynomials in q and p respectively satisfying the recurrence relations:

$$A_{n+1} = q^n + p \sum_{\ell=r(n+1)}^{n} A_\ell q^{n-\ell} \quad , \; A_1 = 1 \qquad \text{(a)}$$

$$B_{n+1} = p^n + q \sum_{\ell=r(n+1)}^{n} B_\ell p^{n-\ell} \quad , \; B_1 = 1 \qquad \text{(b)}$$

Hence, both A_n and B_n have integral coefficient and are of degree less than n.

<u>Proof of Proprosition 1</u>

Let P be the transition matrix for the chain f_n, where first n rows and columns correspond to states (1,h),(2,h), ...,(n,h) and the following n rows and columns to states (n,t),(n-1,t),...,(1,t).

Let $\bar{\mu} = (\mu(1,h),\ldots,\mu(n,h),\mu(n,t),\ldots,\mu(1,t))$ be the stationary distribution, so that

$$\bar{\mu}(I - P) = \bar{0} , \qquad (1)$$

where I is the identity matrix. Partition the matrix P into four submatrices in the form

$$P = \begin{bmatrix} P_h & Q_h \\ Q_t & P_t \end{bmatrix} ,$$

where P_h is an nxn matrix of the form

$$
\begin{array}{rcl}
 & & \text{row:} \\
P_h = & \left[\begin{array}{ccccccccc}
p & q & & & & & & & 0 \\
p & 0 & q & & & \text{All 0's} & & & 0 \\
 & & & \cdot & & & & & \vdots \\
 & p & & 0 & q & & & & 0 \\
 & & & & \cdot & & & & \cdot \\
 & & & p & & 0 & q & & 0 \\
 & & & & & & \cdot & & \cdot \\
 & & & & & & & \cdot & \\
 & & & & & & & & q \\
 & & & p & & & & & 0
\end{array}\right] & \begin{array}{l} 1 \\ 2 \\ \\ r(i) \\ \\ i \\ \\ \\ \\ n \end{array} \\
\text{column:} & \begin{array}{ccccccccc} 1 & 2 & & r(i) & & & i & & n \end{array} &
\end{array}
$$

and P_t is an nxn matrix of the form

$$
\begin{array}{rcl}
 & & \text{row:} \\
P_t = & \left[\begin{array}{cccccccc}
0 & & & & q & & & \\
p & & \cdot & & & \cdot & & \\
0 & p & 0 & & & q & & \\
 & & & & & & & \\
p & & & p & 0 & q & & \\
 & & & & & & & \\
0 & & & & & p & 0 & q \\
0 & & & & & & p & q
\end{array}\right] & \begin{array}{l} n \\ \\ i \\ \\ r(i) \\ \\ 2 \\ 1 \end{array} \\
\text{column:} & \begin{array}{cccccccc} n & & & i & & r(i) & 2 & 1 \end{array} &
\end{array}
$$

According to the sequence $\underline{r}$, these 2 matrices will have all entries located in the same form except that they are opposite to each other in both position and p,q values.

Notice that each row of these matrices contains exactly one entry q namely the $(i,r(i))^{\underline{th}}$ one, and that the labelling of rows and columns of P_t begins at the lower right corner while the labelling of P_h begins from the upper left corner. The off-diagonal matrices Q_h and Q_t consist of all zeros except for the lower left corner of Q_h which is q, and the upper right corner entry of Q_t, which is p.

With this partition, equation (1) decomposes in to two equations

$$\bar{\mu}_h(I - P_h) = (0,0,\ldots,0,\mu(n,t)p) \tag{2}$$

$$\bar{\mu}_t(I - P_t) = (\mu(n,h)q,0,\ldots,0), \tag{3}$$

where

$$\bar{\mu}_h = (\mu(1,h),\ldots,\mu(n,h))$$

$$\bar{\mu}_t = (\mu(n,t),\ldots,\mu(1,t)).$$

From (2) we have $\bar{\mu}_h = (0,0,\ldots,0,\mu(n,t)p)(I - P_h)^{-1}$.

Let

$$\alpha_{ij} = (i,j)^{\underline{th}} \text{ entry of } (I - P_h)^{-1}$$

$$\therefore \quad \mu(i,h) = \mu(n,t)p\alpha_{n,i} \quad , \; i = 1,2,\ldots,n. \tag{4}$$

Using the formula for matrix inversion we get

$$\alpha_{ij} = \frac{(I - P_h)_{(j,i)}}{|I - P_h|}$$

where $|I - P_h|$ is the determinant of $(I - P_h)$ and $(I - P_h)_{(j,i)}$ is the $(j,i)^{\underline{th}}$ cofactor of $(I - P_h)$ matrix. Hence

$$\mu(S_h) = \sum_{i=1}^{n} \mu(i,h) = \frac{\mu(n,t)p}{|I - P_h|} \sum_{i=1}^{n} (I - P_h)_{(i,n)}. \quad (5)$$

Let A_n be the determinant of the nxn matrix obtained from $(I - P_h)$ after replacing the $n^{\underline{th}}$ column of $(I - P_h)$ matrix by a column of 1's,

$$A_n = \begin{vmatrix} q & -q & & & & & & 1 \\ -p & 1 & -q & & & & & 1 \\ & & \ddots & & & & & \vdots \\ & -p & & \ddots & & & & \vdots \\ & & & 1 & -q & & & 1 \\ & & & & \ddots & & & \vdots \\ & & -p & & & 1 & -q & 1 \\ & & & & & & \ddots & \vdots \\ & -p & & & & & & 1 \end{vmatrix} \quad (6)$$

Expanding A_n along this last column we obtain

$$A_n = \sum_{i=1}^{n} (I - P_h)_{(i,n)},$$

since the $(i,n)^{\underline{th}}$ cofactors of A_n and $(I - P_h)$ are identical thus $\mu(S_h)$ in (5) can be rewritten as

$$\mu(S_h) = \frac{\mu(n,t)p}{|I - P_h|} A_n.$$

Since the only transition between S_h and S_t is through state (n,h) and (n,t) we must have

$$\mu(n,t)p = \mu(n,h)q \quad (7)$$

in the stationary regime.

Using the identity in (4) we get

$$\mu(n,h) = \mu(n,t)p\frac{(I - P_h)_{(n,n)}}{|I - P_h|}$$

and substituting in (7) we get

$$|I - P_h| = q(I - P_h)_{(n,n)}.$$

Consider $(I - P_h)_{(n,n)}$ which has the form

$$(I - P_h)_{(n,n)} = \begin{vmatrix} q & -q & & & & & & \\ -p & 1 & -q & & & & & \\ & & \ddots & & & & & \\ & -p & & 1 & -q & & & \\ & & & & \ddots & & & \\ & & & -p & & 1 & -q & \\ & & & & & & \ddots & -q \\ & & -p & & & & & 1 \end{vmatrix}$$

which is the same as the determinant of the $(n-1)x(n-1)$ matrix $I - P_h$ obtained for the chain f_{n-1}. Putting temporarily the superscript (n) for the number of states in S_h, we have a recurrence relation

$$|I^{(n)} - P_h^{(n)}| = q| I^{(n-1)} - P_h^{(n-1)}|$$

Since

$$|I^{(1)} - P_h^{(1)}| = q$$

we obtain

$$|I^{(n)} - P_h^{(n)}| = q^n.$$

Thus

$$\mu(S_h) = \frac{A_n}{q^n}\,\mu(n,t)p.$$

Going back to (3) and repeating all the steps found above we can end up with the similar expression

$$\mu(S_t) = \frac{B_n}{p^n}\,\mu(n,h)q$$

where B_n is the determinant of order n

$$B_n = \begin{vmatrix} 1 & & & & & -q & & \\ \vdots & \ddots & & & & & & \\ \vdots & & \ddots & -q & & & & \\ 1 & -p & 1 & & & & & \\ \vdots & & & \ddots & & & & \\ \vdots & & & -p & 1 & -q & & \\ \vdots & & & & & \ddots & & \\ \vdots & & & & & & 1 & -q \\ 1 & & & & & & -p & -p \end{vmatrix}$$

B_n is the determinant of the matrix $(I - P_t)$ after we replace its $n^{\underline{th}}$ column (in the most left hand side) by a column of 1's.

Hence, using (7) again, we have

$$\frac{\mu(S_h)}{\mu(S_t)} = \frac{p^n}{q^n}\,\frac{A_n}{B_n}$$

and it left to prove that A_n and B_n have the form as stated in (a) and (b).

Consider A_n first. To evaluate this determinant let

$$I_1 = \{i = 2,3,\ldots,n:\ r(i) = 1\}.$$

Notice that I_1 is the set of exactly those rows indices i for which the $(i,1)^{\underline{th}}$ entries in (6) are -p. Multiply the first row in (6) by p/q and add it to all rows such that $i\varepsilon I_1$. This operation does not change the determinant of the matrix. So we have

$$A_n = \left|\begin{array}{cccccccc}
q & -q & & & & & & t_{21}^{(n)} \\
0 & q & -q & & & & & \cdot \\
\cdot & -p & 1 & -q & & & & \cdot \\
\cdot & & \cdot & & & & & \cdot \\
\cdot & & & \cdot & & & & \cdot \\
 & & & 1 & -q & & & \\
0 & & & & \cdot & & & \cdot \\
\cdot & & & & & \cdot & & \cdot \\
\cdot & & -p & & & 1 & -q & \\
\cdot & & & & & & \cdot & \cdot \\
\cdot & & & & & & \cdot & \\
\cdot & & & & & & \cdot & \\
0 & & & & & & & t_{2n}^{(n)}
\end{array}\right|$$

The entries in the last column are given by

$$t_{21}^{(n)} = \begin{cases} 1 + \frac{p}{q} & \text{if } i\varepsilon I_1 \\ 1 & \text{if } i\not\varepsilon I_1 \end{cases}$$

Expanding this determinant along the first column we have

$$D_2^{(n)} = qD_1^{(n)} ,$$

where

$D_1^{(n)} = A_n$ is the original determinant (6) and

$$D_2^{(n)} = \begin{vmatrix} q & -q & & & & & & t_{22}^{(n)} \\ -p & 1 & -q & & & & & \cdot \\ & & \ddots & & & & & \cdot \\ & & & 1 & -q & & & \cdot \\ & -p & & & \ddots & & & \cdot \\ & & & & & 1 & -q & \cdot \\ & & & -p & & & \ddots & \cdot \\ & & & & & & & t_{2n}^{(n)} \end{vmatrix}$$

This determinant is of order n-1. Notice that the entries in the first column in the determinant $D_2^{(n)}$ are -p only for row indices 1 = 3,...,n such that either r(i) = 1 or r(i) = 2. Hence, letting $I_2 = \{i = 3,4,\ldots,n\colon r(i) < 3\}$, multiplying the first row in $D_2^{(n)}$ by p/q and adding to rows with $i\varepsilon I_2$ this determinant becomes

$$\begin{vmatrix} q & -q & & & & & t_{32}^{(n)} \\ 0 & q & -q & & & & \cdot \\ \cdot & & \ddots & & & & \cdot \\ \cdot & & & 1 & -q & & \cdot \\ \cdot & & & & \ddots & & \cdot \\ \cdot & -p & & & 1 & -q & \cdot \\ \cdot & & & & & \ddots & \cdot \\ 0 & & & & & & t_{3n}^{(n)} \end{vmatrix}$$

The entries in the last column are

$$t_{3i}^{(n)} = \begin{cases} t_{2i}^{(n)} + \frac{q}{p}\, t_{22}^{(n)} & \text{if } i \varepsilon I_2 \\ \\ t_{2i}^{(n)} & \text{if } i \notin I_2 \end{cases}$$

Expanding again along the first column we have

$$D_2^{(n)} = qD_3^{(n)} \ ,$$

where

$$D_3^{(n)} = \begin{vmatrix} q & -q & & & & & t_{33}^{(n)} \\ \vdots & \ddots & \ddots & & & & \vdots \\ \cdot & & 1 & -q & & & \vdots \\ & & & \ddots & \ddots & & \vdots \\ & -p & & & 1 & -q & \vdots \\ \cdot & & & & & \ddots & \vdots \\ & & & & & & t_{3n}^{(n)} \end{vmatrix}$$

Proceeding in this fashion we obtain a sequence of determinant $D_1^{(n)}, D_2^{(n)}, \ldots, D_n^{(n)}$ (8)

where determinate $D_k^{(n)}$ is of order n-k+1. The entries $t_{ki}^{(n)}$ in the last column of $D_k^{(n)}$ satisfy the recurrence relation

$$t_{k+1,i}^{(n)} = \begin{cases} t_{ki}^{(n)} + \frac{q}{p}\, t_{k,k}^{(n)} & \text{if } i \varepsilon I_k \\ \\ t_{ki}^{(n)} & \text{if } i \notin I_k \ , \end{cases} \qquad (9)$$

$k = 1,2,\ldots,n;\ i = k,k+1,\ldots,n;$ and $t_{1,i}^{(n)} = 1$, where

$$I_k = \{i = k+1, k+2, \ldots, n:\ \ r(i) \leq k\}.$$

Further

$$D_{k+1}^{(n)} = qD_k^{(n)},$$

so that

$$D_1^{(n)} = q^{n-1}D_n^{(n)} = q^{n-1}t_{n,n}^{(n)} \,. \tag{10}$$

Consider the determinant A_{n+1} of order n+1 obtained by using the same sequence $\underline{r}$,

$$A_{n+1} = \begin{vmatrix} q & -q & & & & & & & & 1 \\ -p & 1 & -q & & & & & & & 1 \\ & \ddots & & & & & & & & \vdots \\ & -p & & 1 & -q & & & & & \vdots \\ & & & & \ddots & & & & & \vdots \\ & & & -p & & 1 & -q & & & 1 \\ & & & & & & \ddots & & & \vdots \\ & & -p & & & & & & 1 & 1 \\ & & & & -p & & & & & 1 \end{vmatrix}$$

Applying the above procedure to $D_1^{(n+1)} = A_{n+1}$ we obtain a sequence

$$D_1^{(n+1)}, \ \ldots \ , D_{n+1}^{(n+1)} \tag{11}$$

where the determinants $D_k^{(n+1)}$ again satisfy (9) with n replaced by n+1. Arrange now the last columns of the sequences (8) and (11) into triangular arrays as follows:

$$T^{(n)} = \begin{cases} t_{1,1}^{(n)} \\ t_{1,2}^{(n)}, \; t_{2,2}^{(n)} \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ t_{1,n}^{(n)}, \; t_{2,n}^{(n)}, \; . \; . \; . \; , t_{n,n}^{(n)} \end{cases}$$

$$T^{(n+1)} = \begin{cases} t_{1,1}^{(n+1)} \\ t_{1,2}^{(n+1)}, \; t_{2,2}^{(n+1)} \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ t_{1,n}^{(n+1)}, \; t_{2,n}^{(n+1)}, \; . \; . \; . \; , t_{n,n}^{(n+1)} \\ t_{1,n+1}^{(n+1)}, \; t_{2,n+1}^{(n+1)}, \; . \; . \; . \; , t_{n,n+1}^{(n+1)}, \; t_{n+1,n+1}^{(n+1)} \end{cases}$$

Since for $i \leq n$ by the definition of sets $I_k^{(n)}$, $i \varepsilon I_k^{(n)}$ if and only if $i \varepsilon I_k^{(n+1)}$, the first n rows of $T^{(n)}$ and $T^{(n+1)}$ are identical, i.e.,

$$t_{k,i}^{(n)} = t_{k,i}^{(n+1)} \quad , \; i = k,\ldots,n; \; k = 1,2,\ldots,n. \tag{12}$$

Next by (9)

$$t_{k,n+1}^{(n+1)} = \begin{cases} 1 & \text{if } k = 1,2,\ldots,r(n+1) \\ 1 + \frac{p}{q} \sum_{\ell=r(n+1)}^{k-1} t_{\ell,\ell}^{(n+1)} & \text{if } k=r(n+1)+1,\ldots,n+1 \end{cases}$$

In particular for k = n+1 since $r(n+1) < n+1$

$$t_{n+1,n+1}^{(n+1)} = 1 + \frac{p}{q} \sum_{\ell=r(n+1)}^{n} t_{\ell,\ell}^{(n+1)} . \tag{13}$$

But by (12) for $\ell < n+1$, we have

$$t_{\ell,\ell}^{(n+1)} = t_{\ell,\ell}^{(n)} = \cdots = t_{\ell,\ell}^{(\ell)} .$$

Hence, from (13) we have

$$t_{n+1,n+1}^{(n+1)} = 1 + \frac{p}{q} \sum_{\ell=r(n+1)}^{n} t_{\ell,\ell}^{(\ell)} .$$

Hence, by (10)

$$\frac{D_1^{(n+1)}}{q^n} = 1 + \frac{p}{q} \sum_{\ell=r(n+1)}^{n} \frac{D_1^{(\ell)}}{q^{\ell-1}} .$$

Since $D_1^{(n)} = A_n$, we have

$$A_{n+1} = q^n + p \sum_{\ell=r(n+1)}^{n} A_\ell q^{n-\ell}, \quad n = 1,2,\ldots \tag{14}$$

where $A_1 = 1$.

The recurrence relation for B_n is established in exactly the same fashion. Notice that the difference of determinants A_n and B_n is only p and q are in reverse position, thus on the same sequence of r we have

$$B_{n+1} = p^n + q \sum_{\ell=r(n+1)}^{n} B_\ell p^{n-\ell}, \quad n = 1,2,\ldots \tag{15}$$

From (14) and (15) we had completed the proof of proprosition 1.

IV. DETERMINATION OF OPTIMAL ALGORITHM FOR SMALL n

Since the error probability is

$$P(e) = \left\{1 + \frac{\mu(S_h)}{\mu(S_t)}\right\}^{-1}$$

in order to minimize P(e) we need to maximize the ratio $\mu(S_h)/\mu(S_t)$.

From the last section, we have

$$\frac{\mu(S_h)}{\mu(S_t)} = \left(\frac{p}{q}\right)^n \frac{A_n}{B_n}$$

where

$$\tfrac{1}{2} < p < 1 \text{ and } q = 1 - p.$$

Thus the ratio $\mu(S_h)/\mu(S_t)$ is maximized if A_n/B_n is maximized. The ratios A_n/B_n depend on the sequence $f_n(\underline{r}) = < r(2), r(3), \ldots, r(n) >$. After we have determined all possible algorithms $f_n(\underline{r})$ (there are $(n-1)!$ possible algorithms in total) and computed the corresponding value of A_n/B_n we can search for the maximum A_n/B_n. This way we can identify the optimal algorithm $f_n^*(\underline{r})$ and obtain the maximum value of $\mu(S_h)/\mu(S_t)$.

In what follows we carry this program for the case of small n. For the case of n = 1, we have P(e) = q which means that the decision made without any memory can be done with probability of error equals to q.

Ratios A_n/B_n were calculated algebraically for n = 2, 3, 4 and 5. For the value of $p \varepsilon (\tfrac{1}{2}, 1)$ it was found that

$$\left(\frac{A_2}{B_2}\right)^* = 1 \qquad \text{for any } f$$

$$\left(\frac{A_3}{B_3}\right)^* = 1 \qquad \text{for any } f$$

$$\left(\frac{A_4}{B_4}\right)^* = \frac{q^3 + p}{p^3 + q} \qquad \text{for } f_4^* = < 1,1,3 >$$

$$\left(\frac{A_5}{B_5}\right)^* = \frac{q^3 + p^2q + p^2}{p^3 + pq^2 + q^2} \qquad \text{for } f_5^* \quad < 1,1,2,4 >$$

The algebraical work to determine these values is contained in Appendix A.

For values of n from 6 to 10, several values of $p > \frac{1}{2}$ were chosen. The search for the optimal algorithm was performed by using an IBM 360/67 in double precision. The program found in Appendix B was becoming too time consuming when n exceeds 10. For the values of p in the vicinity of 1, value of p at .99 and .999 were used. However, in the vicinity of $\frac{1}{2}$, the optimal algorithm was determined by a Taylor series expansion around $\frac{1}{2} + \varepsilon$, and neglecting terms with ε having power greater than or equal to 2. The derivation of this expansion was as follows:

$$A_n(q) = A_n(\tfrac{1}{2} - \varepsilon) = A_n(\tfrac{1}{2}) - \varepsilon A_n'(\tfrac{1}{2}) + \frac{A_n''(\frac{1}{2})}{2!}(-\varepsilon)^2 + R$$

where R is the remainder term.

Ignoring the terms which have ε^n where $n \geq 2$, we get $A_n(\frac{1}{2} - \varepsilon) \cong A_n(\frac{1}{2}) - \varepsilon A_n'(\frac{1}{2})$. Similarly $B_n(\frac{1}{2} + \varepsilon) \cong A_n(\frac{1}{2}) + \varepsilon A_n'(\frac{1}{2})$. Consider

$$X = \frac{A_n(q)}{B_n(p)} \cong \frac{A_n - \varepsilon A_n'}{A_n + \varepsilon A_n'}$$

$$= \frac{1 - \varepsilon \frac{A_n'}{A_n}}{1 + \varepsilon \frac{A_n'}{A_n}} .$$

To maximize X, we minimize A_n'/A_n. Differentiating A_n with respect to q we get

$$A_{k+1}'(q) = kq^{k-1} + \sum_{\ell=r(k+1)}^{k} q^{k-\ell+1} A_\ell'(q) + \sum_{\ell=r(k+1)}^{k} (k-\ell-1) A_\ell(q) q^{k-\ell}$$

which is recursive in A_k' and A_k, where $q = \frac{1}{2}$.

The program for the search of the ratio A_n'/A_n is at Appendix C. Notice that this approximation of p value yield the same result in term of optimal algorithm as in the case when p = .51 shown in Table I and II.

These results are summarized in Table I, Table II and Figure 5, 6 and 7. Table II presents the results of Table I in the form used in [5] by Anderson. The result was not so much different although, the algorithm presented here has no stopping rule. After some amount of observation were received the decisions are made with the same probability of error as in [5]. Anderson in his paper made a point that the finite-memory algorithm seems to operate in a similar fashion as a human decision making process. The result of this paper seems to support his observation. The mechanism of remembering and forgetting seems to resemble somewhat the

procedure of making decision under finite memory constraint. The fact that man can gain experience and learn during his entire life span, this makes one to believe that there must be some kind of data summarized mechanism such that he can make a decision with some degree of confidence at a given moment. The word experience may be one of the key which can lead to further study about the mechanism of data summarization. The algorithm proposed in this paper can be viewed as a data summarized procedure under simple hypotheses with Bernoulli observations. The more complicated machine can be developed by considering the general case of multiple hypotheses testing under finite state memory. The idea was discussed in [7] by Yakowitz and in [9] by Salagowicz.

p	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
0.50	∀p	∀p	∀p	∀p	∀p	<1,1,1,3,5,6>	<1,1,1,2,4,6,7>	<1,1,1,2,4,6,7,8>	<1,1,1,3,4,5,7,8,9>
.55						"	"	"	<1,1,1,2,3,5,7,8,9>
.60						"	"	<1,1,1,3,4,5,7,8>	"
.65						<1,1,1,2,4,6>	"	<1,1,1,2,3,5,7,8>	<1,1,1,2,4,5,6,8,9>
.70		<1,1>				"	<1,1,1,3,4,5,7>	"	<1,1,1,2,3,4,6,8,9>
.75	<1>	OR	<1,1,3>	<1,1,2,4>	<1,1,1,3,5>	"	<1,1,1,2,3,5,7>	<1,1,1,2,4,5,6,8>	<1,1,1,2,3,5,6,7,9>
.80		<1,2>				"	"	<1,1,1,2,3,4,6,8>	<1,1,1,2,3,4,5,7,9>
.85						"	"	<1,1,1,1,3,4,6,8>	<1,1,1,1,2,4,5,7,9>
.90						"	"	"	"
.95						"	"	"	"
.99						"	"	"	"
.999						<1,1,1,1,4,6>	<1,1,1,1,1,5,7>	<1,1,1,1,1,1,6,8>	<1,1,1,1,1,1,1,7,9>

Table I.

Optimal Algorithm $f^* = < r(2), \dots ,r(n) >, r(i)$ = State that the Transition Goes from i.

p	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
0.50	∀p	∀p	∀p	∀p	∀p	<1,2,3,2,1,1>	<1,2,3,3,2,1,1>	<1,2,3,3,2,1,1,1>	<1,2,3,2,2,2,1,1,1>
.55						"	"	"	<1,2,3,3,3,2,1,1,1>
.60						"	"	<1,2,3,2,2,2,1,1>	"
.65						<1,2,3,3,2,1>	"	<1,2,3,3,3,2,1,1>	<1,2,3,3,2,2,2,1,1>
.70		<1,1>				"	<1,2,3,2,2,2,1>	"	<1,2,3,3,3,3,2,1,1>
.75	<1>	OR	<1,2,1>	<1,2,2,1>	<1,2,3,2,1>	"	<1,2,3,3,3,2,1>	<1,2,3,3,2,2,2,1>	<1,2,3,3,3,2,2,2,1>
.80		<1,2>				"	"	<1,2,3,3,3,3,2,1>	<1,2,3,3,3,3,3,2,1>
.85						"	"	<1,2,3,4,3,3,2,1>	<1,2,3,4,4,3,3,2,1>
.90						"	"	"	"
.95						"	"	"	"
.99						"	"	"	"
.999						<1,2,3,4,2,1>	<1,2,3,4,5,2,1>	<1,2,3,4,5,6,2,1>	<1,2,3,4,5,6,7,2,1>

Table II.

Optimal Algorithm $f^* = < \rho(2),\ldots,\rho(n) >$ $\rho(i)$ = Number of States that the Transition Transits From State i.

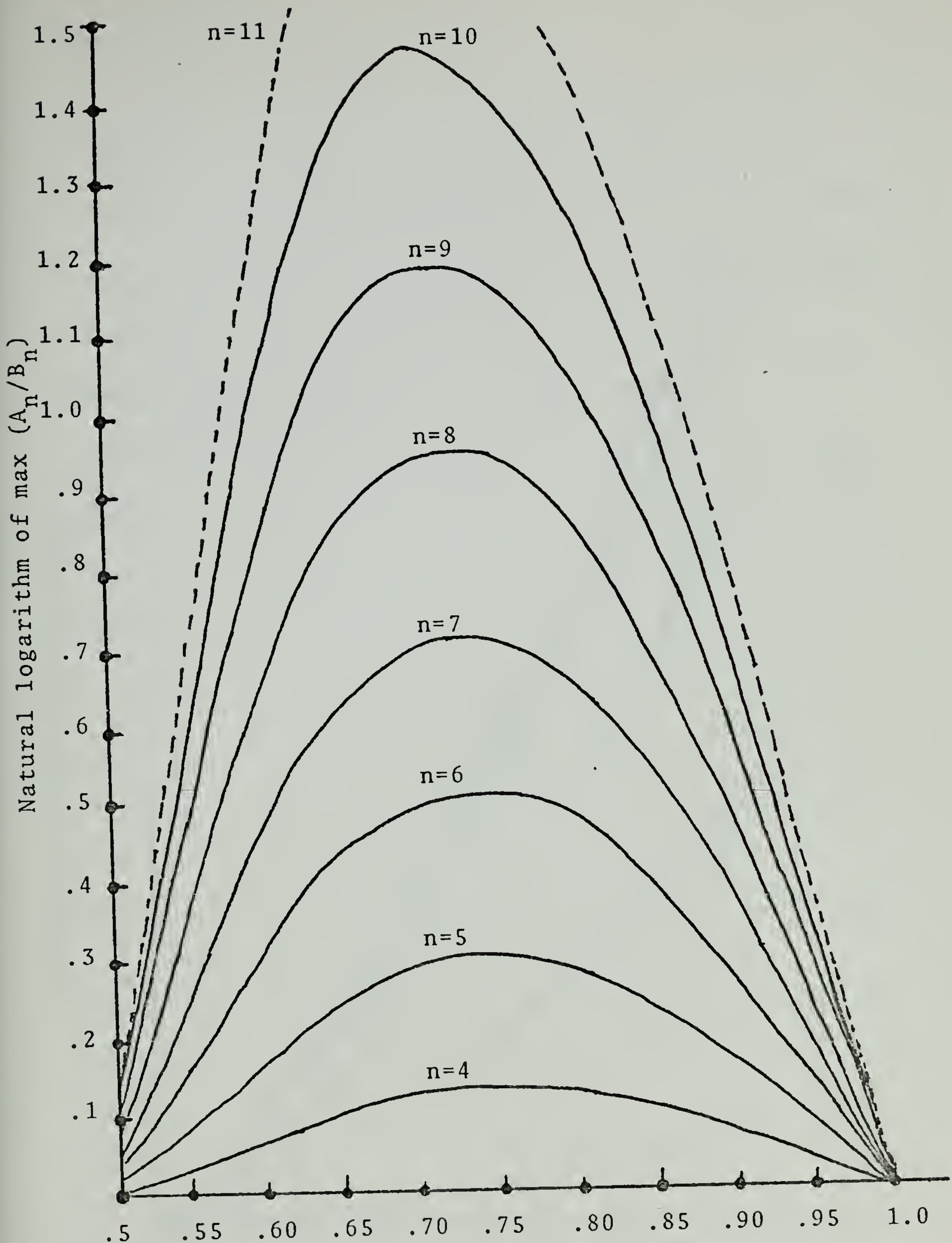


Figure 5. Graph of $\ell n \max \frac{A_n}{B_n}$ vs. p values.

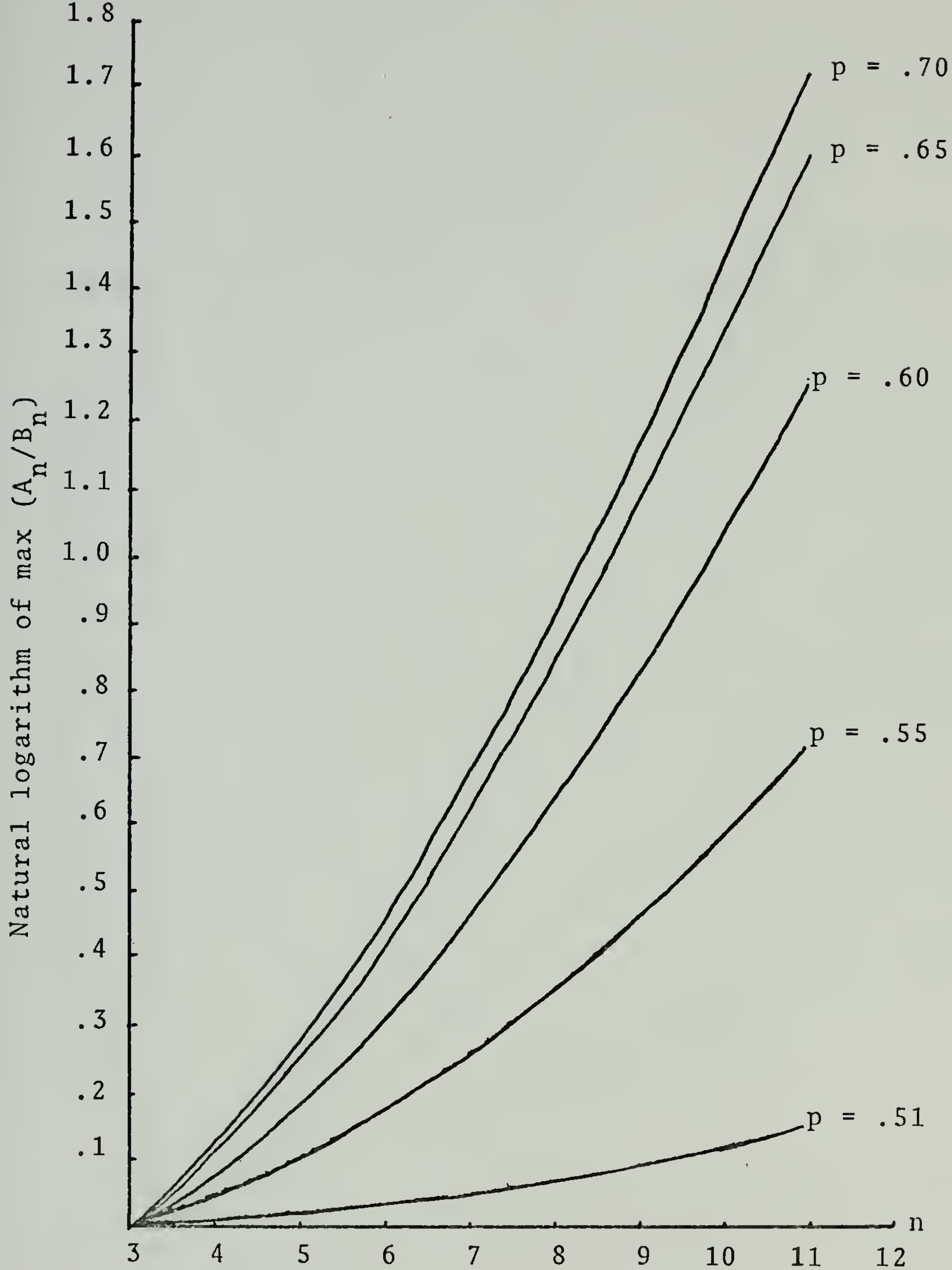


Figure 6. Graph of ℓn max (A_n/B_n) vs. n values (p from .50-.70)

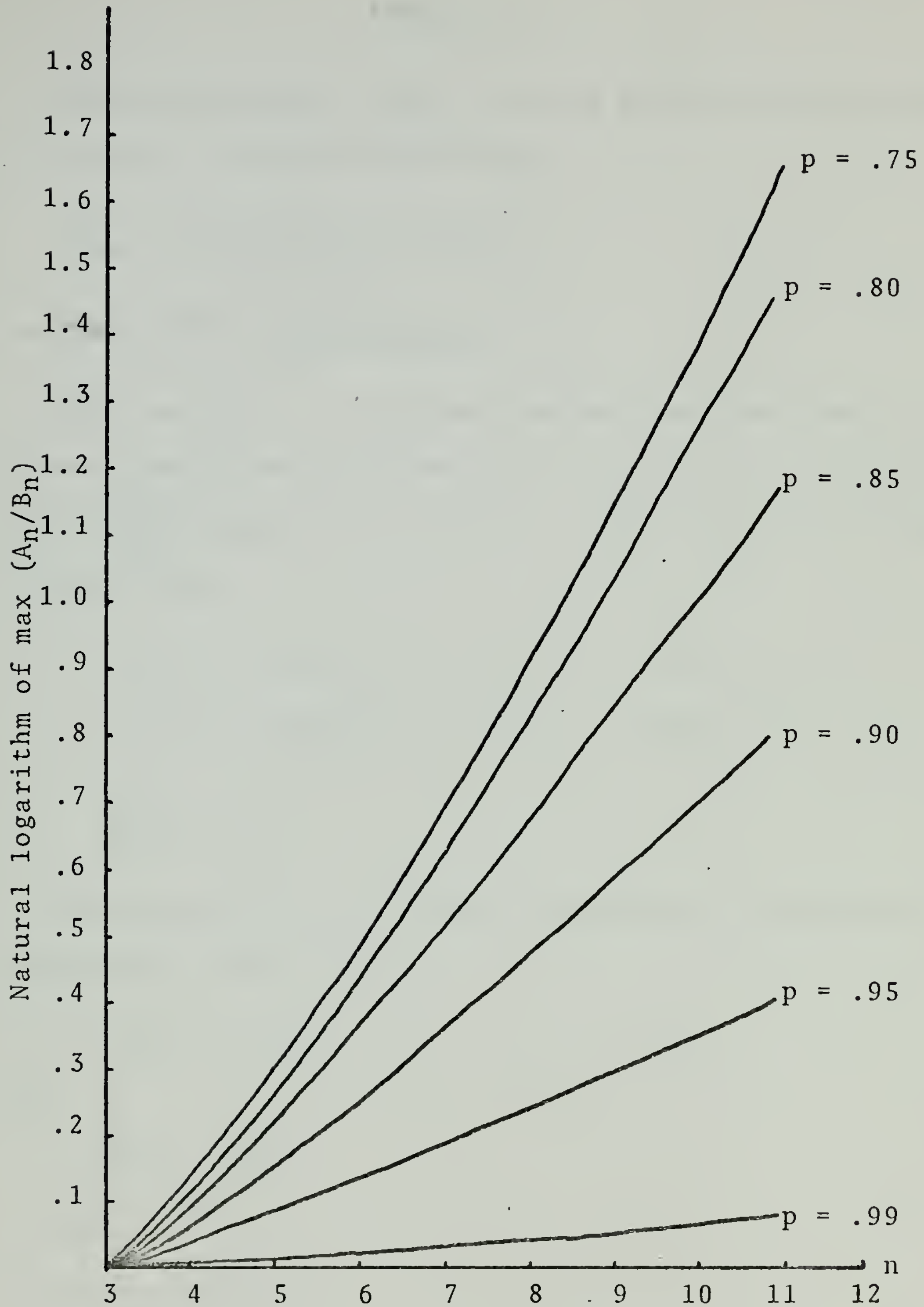


Figure 7. Graph of ℓn max (A_n/B_n) vs. n values (p from .75-.99)

APPENDIX A

Determination of $(A_n/B_n)^*$ and f_n^* algebraically for n=2, 3, 4, and 5. Recall that we have

$$A_{k+1} = q^k + p\sum_{\ell=r(k+1)}^{k} A_\ell q^{k-\ell}$$

$$B_{k+1} = p^k + q\sum_{\ell=r(k+1)}^{k} B_\ell p^{k-\ell}.$$

The case of n = 2, we have the unique algorithm f = <1> because the transition from state 2 to state 1 can be done only by the jump of 1 or r(2) = 1. According to the formula above we have

$$A_2 = q + p(A_1 q^o) = q + p = 1 \quad \text{where } A_1 = 1$$

$$B_2 = p + q(B_1 p^o) = p + q = 1 \quad \text{where } B_1 = 1$$

$$\frac{A_2}{B_2} = 1.$$

The case of n = 3, we have two possible algorithms to investigate, namely,

$$f_3^1 = < 1,1 >$$

and

$$f_3^2 = < 1,2 >.$$

For

$$f_3^1 = < 1,1 >;$$

$$A_3 = q^2 + p(A_1 q + A_2 q^o)$$

$$A_2 = 1$$

$$\therefore \; A_3 = q^2 + pq + p$$

$$= q + p$$

$$= 1.$$

By the same derivation,

$$B_3 = p^2 + pq + q$$

$$= p + q$$

$$= 1$$

$$\therefore \left(\frac{A_3}{B_3}\right)^1 = 1.$$

For

$$f_3^2 = < 1,2 >;$$

$$A_3 = q^2 + pA_2$$

$$= q^2 + p$$

and

$$B_3 = p^2 + q.$$

But

$$p^2 + q = q^2 + p \qquad \text{for } \tfrac{1}{2} < p = 1 - q < 1$$

$$\therefore \left(\frac{A_3}{B_3}\right)^2 = 1.$$

Note that, the superscripts at the ratio are for numbering purposes.

Thus for the case n = 3, the optimal algorithm can be either $f_3^* = < 1,1 >$ or $f_3^* = < 1,2 >$.

Before we proceed the proof for the case n = 4 and 5, we note that the expression for A_n and B_n differ only in p and q being interchanged. For this reason, the expressions for B_n will be presented without computation.

Determination of A_n/B_n and f_n^* algebraically for n = 4 and n = 5. Recall that

$$A_{k+1} = q^k + p\sum_{\ell=r(k+1)}^{k} A_\ell q^{k-\ell}$$

$$B_{k+1} = p^k + q\sum_{\ell=r(k+1)}^{k} B_\ell p^{k+\ell}.$$

For the case when n = 4, the possible forms of algorithm are (n - 1) ! = 3 ! = 6, which are as follows

$$f_4^1 = \langle 1,1,1 \rangle,\ f_4^2 = \langle 1,1,2 \rangle,\ f_4^3 = \langle 1,1,3 \rangle$$

$$f_4^4 = \langle 1,2,1 \rangle,\ f_4^5 = \langle 1,2,2 \rangle,\ f_4^6 = \langle 1,2,3 \rangle.$$

The optimal algorithm among these combinations is $f_4^* = f_4^3$ and $(A_4/B_4)^* = (A_4/B_4)^3 = (q^3+p)/(p^3+q)$ where superscript labels the combination order.

<u>Proof</u>

For f_4^1;

$$A_4 = q^3 + p(A_1q^2 + A_2q + A_3)$$

$$A_3 = q^2 + p(A_1q + A_2)$$

$$A_2 = q + pA_1$$

$$A_1 = 1$$

$$\therefore A_2 = 1$$

and

$$A_3 = q^2 + pq + p = q + p = 1$$

hence

$$A_4 = q^3 + pq^2 + pq + p$$

$$= q^2 (q + p) + pq + p$$

$$= q^2 + pq + p = 1.$$

The same result can be obtained from computing B_4

$$\therefore \left(\frac{A_4}{B_4}\right)^1 = 1.$$

It was proved that the case when $n = 2$, the transition always goes to 1. So the value of A_2 always equals to 1 so as B_2.

For $f_4^2 = < 1,1,2 >$;

$$A_4 = q^3 + p(A_2q + A_3)$$

$$A_3 = q^2 + p(A_1q + A_2)$$

where

$$A_1 = A_2 = 1$$

$$= q^2 + pq + p = 1$$

$$\therefore A_4 = q^3 + pq + p$$

$$B_4 = p^3 + pq + q$$

$$\left(\frac{A_4}{B_4}\right)^2 = \frac{q^3 + pq + p}{p^3 + pq + q}.$$

For $f_4^3 = < 1,1,3 >$;

$$A_4 = q^3 + pA_3$$

$$A_3 = 1$$

$$\therefore \quad A_4 = q^3 + p$$

$$\therefore \left(\frac{A_4}{B_4}\right)^3 = \frac{q^3 + p}{p^3 + q} .$$

For $f_4^4 = < 1,2,1 >$;

$$A_4 = q^3 + p(A_1q^2 + A_2q + A_3)$$

$$A_3 = q^2 + pA_2$$

$$= q^2 + p$$

$$\therefore \quad A_4 = q^3 + 2pq^2 + p = q^2 + pq^2 + p$$

$$\therefore \left(\frac{A_4}{B_4}\right)^4 = \frac{q^3 + 2pq^2 + p}{p^3 + 2pq^2 + q} .$$

For $f_4^5 = < 1,2,2 >$;

$$A_4 = q^3 + p(A_2q + A_3)$$

$$A_3 = q^2 + p$$

hence

$$A_4 = q^3 + pq + pq^2 + p^2 = q^2 + p$$

$$\therefore \left(\frac{A_4}{B_4}\right)^5 = \frac{q^2 + p}{p^2 + q} = 1 \quad ,$$

since

$$q^2 + p = p^2 + q.$$

For $f_4^6 = < 1,2,3 >$;

$$A_4 = q^3 + pA_3$$

$$A_3 = q^2 + p$$

$$\therefore\ A_4 = q^3 + pq^2 + p^2$$

$$= q^2 + p^2$$

$$\therefore \left(\frac{A_4}{B_4}\right)^6 = 1.$$

Consider $1 > p$ where $p > \frac{1}{2}$.

$$p > p^2$$

$$1 > p^2 - p + 1$$

$$1 > p^2 + q = p^2 + pq + q^2$$

$$p - q > (p - q)(P^2 + pq + q^2) = p^3 - q^3$$

$$q^3 + p > p^3 + q$$

$$\frac{q^3 + p}{p^3 + q} > 1$$

or $(A_4/B_4)^3 > 1$ and $(A_4/B_4)^1$, $(A_4/B_4)^5$ and $(A_4/B_4)^6$ are dominated.

Stated without a proof is the following simple fact;

Let $\quad a > 0$; $b > 0$ and $c > 0$

if $\quad a/b > 1$ then $a/b > a + c/b + c$.

This result will be applied when needed without notice.

$$\therefore\ \frac{q^3 + p}{p^3 + q} > \frac{q^3 + pq + p}{p^3 + pq + q}$$

$$\therefore\ \left(\frac{A_4}{B_4}\right)^3 > \left(\frac{A_4}{B_4}\right)^2 .$$

Obviously,

$$\left(\frac{A_4}{B_4}\right)^3 > \left(\frac{A_4}{B_4}\right)^4$$

since

$$\frac{q^3 + p}{p^3 + q} > \frac{q^3 + 2pq^2 + p}{p^3 + 2p^2q + q}$$

where

$$2pq^2 < 2p^2q.$$

Thus for $n = 4$, the optimal algorithm is $f_4^* = < 1,1,3 >$.

For the case of $n = 5$, the possible forms of algorithms $f = < r(2), r(3), r(4), r(5) >$ are as follows:

$f_5^1 = < 1,1,1,1 >$ $\quad$ $f_5^2 = < 1,1,1,2 >$

$f_5^3 = < 1,1,1,3 >$ $\quad$ $f_5^4 = < 1,1,1,4 >$

$f_5^5 = < 1,1,2,1 >$ $\quad$ $f_5^6 = < 1,1,2,2 >$

$f_5^7 = < 1,1,2,3 >$ $\quad$ $f_5^8 = < 1,1,2,4 >$

$f_5^9 = < 1,1,3,1 >$ $\quad$ $f_5^{10} = < 1,1,3,2 >$

$f_5^{11} = < 1,1,3,3 >$ $\quad$ $f_5^{12} = < 1,1,3,4 >$

$f_5^{13} = < 1,2,1,1 >$ $\quad$ $f_5^{14} = < 1,2,1,2 >$

$f_5^{15} = < 1,2,1,3 >$ $\quad$ $f_5^{16} = < 1,2,1,4 >$

$f_5^{17} = < 1,2,2,1 >$ $\quad$ $f_5^{18} = < 1,2,2,2 >$

$f_5^{19} = < 1,2,2,3 >$ $\quad$ $f_5^{20} = < 1,2,2,4 >$

$f_5^{21} = < 1,2,3,1 >$ $\quad$ $f_5^{22} = < 1,2,3,2 >$

$f_5^{23} = < 1,2,3,3 >$ $\quad$ $f_5^{24} = < 1,2,3,4 >$.

The following algebraical proof reveals that the optimal algorithm is $f_5^* = f_5^8 = < 1,1,2,4 >$ and the maximum ratio A_5/B_5 of this algorithm is $(q^3 + 2p^2q + p^3)/(p^3 + 2pq^2 + q^3)$ for all value of $p \varepsilon (\frac{1}{2},1)$.

<u>Proof</u>. If it is simpler to break the possible f's into a group of 4 and identify the maximum ratio A_5/B_5 among the 4 and the result of these 6 groups will be compared later.

$f_5^1 = < 1,1,1,1 >$; from the case n = 4, $f_4 = < 1,1,1 >$ we have

$$A_1 = 1,\ A_2 = 1,\ A_3 = 1,\ A_4 = 1.$$

$$\begin{aligned}
\therefore A_5 &= q^4 + p(A_1q^3 + A_2q^2 + A_3q + A_4) \\
&= q^4 + pq^3 + pq^2 + pq + p \\
&= 1 \\
&= B_4
\end{aligned}$$

$$\therefore \left(\frac{A_5}{B_5}\right)^1 = 1$$

$$f_5^2 = < 1,1,1,2 >;$$

$$\begin{aligned}
A_5 &= q^4 + p(A_2q^2 + A_3q + A_4) \\
&= q^4 + pq^2 + pq + p
\end{aligned}$$

$$\therefore B_5 = p^4 + p^2q + pq + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^2 = \frac{q^4 + pq^2 + pq + p}{p^4 + p^2q + pq + q}$$

$$f_5^3 = < 1,1,1,3 >;$$

$$A_5 = q^4 + p(A_3 q + A_4)$$
$$= q^4 + pq + p$$

and

$$B_5 = p^4 + pq + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^3 = \frac{q^4 + pq + p}{p^4 + pq + q}$$

$$f_5^4 = < 1,1,1,4 >;$$

$$A_5 = q^4 + pA_4$$
$$= q^4 + p$$

$$B_5 = p^4 + q$$

$$\left(\frac{A_5}{B_5}\right)^4 = \frac{q^4 + p}{p^4 + q} .$$

Consider $2pq > 0 \quad \forall p \varepsilon (\tfrac{1}{2}, 1)$

$$1 + 2pq > 1$$
$$1 > (p + q)^2 - 2pq$$
$$1 > p^2 + q^2 = p^2 + pq^2 + q^3$$
$$= p(p + q^2) + q^3$$
$$= p(p^2 + q) + q^3$$
$$= p^3 + pq + q^3$$
$$= \frac{p^4 - q^4}{p - q}$$

$$\therefore \quad p - q > p^4 - q^4$$

$$q^4 + p > p^4 + q$$

$$\frac{q^4 + p}{p^4 + q} = \left(\frac{A_5}{B_5}\right)^4 > 1$$

$$\therefore \left(\frac{A_5}{B_5}\right)^4 > \left(\frac{A_5}{B_5}\right)^1$$

and

$$\frac{q^4 + p}{p^4 + q} > \frac{(q^4 + p) + pq}{(p^4 + q) + pq} \qquad \left(\frac{A_5}{B_5}\right)^4 > \left(\frac{A_5}{B_5}\right)^3$$

and

$$\frac{q^4 + p}{p^4 + q} > \frac{(q^4 + p) + pq^2 + pq}{(p^4 + q) + p^2q + pq} \quad \text{since } pq^2 < p^2q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^4 > \left(\frac{A_5}{B_5}\right)^2$$

$$\therefore \left(\frac{A_5}{B_5}\right)^4 \text{ is maximum in this group of 4.}$$

$f_5^5 = < 1,1,2,1 >$; from the case $n = 4$, the algorithm of the form $f_3 = < 1,1,2 >$ we have $A_1 = A_2 = A_3 = 1$ and $A_4 = q^3 + pq + p$.

$$\begin{aligned} A_5 &= q^4 + p(A_1q^3 + A_2q^2 + A_3q + A_4) \\ &= q^4 + pq^3 + pq^2 + pq + pq^3 + p^2q + p^2. \end{aligned}$$

After simplifying we get

$$A_5 = (p^2 + q)(pq + 1)$$

$$B_5 = (q^2 + p)(pq + 1)$$

$$\left(\frac{A_5}{B_5}\right)^5 = \frac{(p^2 + q)(pq + 1)}{(q^2 + p)(pq + 1)} = 1 \text{ since } p^2 + q = q^2 + p.$$

$$f_5^6 = < 1,1,2,2 > ;$$

$$A_5 = q^4 + p(A_2q^2 + A_3q + A_4)$$

$$= q^4 + pq^2 + pq + pq^3 + p^2q + p^2$$

$$= q^2 + p^2q + p$$

$$\therefore B_5 = p^2 + pq^2 + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^6 = \frac{q^2 + p^2q + p}{p^2 + pq^2 + q} .$$

$$f_5^7 = \langle 1,1,2,3 \rangle;$$

$$A_5 = q^4 + p(A_3q + A_4)$$

$$= q^4 + pq + pq^3 + p^2q + p^2$$

$$= q^3 + p^2q + p$$

$$B_5 = p^3 + pq^2 + q$$

and

$$\left(\frac{A_5}{B_5}\right)^7 = \frac{q^3 + p^2q + p}{p^3 + pq^2 + q} .$$

$$f_5^8 = \langle 1,1,2,4 \rangle;$$

$$A_5 = q^4 + p(A_4)$$

$$= q^4 + pq^3 + p^2q + p^2$$

$$= q^3 + p^2q + p^2$$

$$\therefore B_5 = p^3 + pq^2 + q^2$$

$$\left(\frac{A_5}{B_5}\right)^8 = \frac{q^3 + p^2q + p^2}{p^3 + pq^2 + q^2}$$

$$= \frac{q^3 + 2p^2q + p^3}{p^3 + 2pq^2 + q^3} .$$

Among 5, 6, 7 and 8, the ratio $(A_5/B_5)^8$ is the maximum.

Consider $2pq > 0$

$$\therefore \quad p(1 - p) + q(1 - q) > 0$$

$$\therefore \quad p - p^2 + q - q^2 > 0 \Longleftrightarrow p + q > p^2 + q^2$$

$$\therefore \quad p + q + pq > p^2 + q^2 + pq$$

$$(p + q)(p - q) + pq(p - q) > (p - q)(p^2 + q^2 + pq)$$

$$p^2 - q^2 + p^2q - pq^2 > p^3 - q^3$$

$$q^3 + p^2q + p^2 > p^3 + pq^2 + q^2 \Rightarrow \left(\frac{A_5}{B_5}\right)^8 > 1 = \left(\frac{A_5}{B_5}\right)^5.$$

Consider

$$\left(\frac{A_5}{B_5}\right)^6 = \frac{q^2 + p^2q + p}{p^2 + pq^2 + q} = \frac{q^3 + p^2q + p^2 + pq^2 + pq}{p^3 + pq^2 + q^2 + p^2q + pq}$$

$$< \frac{q^3 + p^2q + p^2}{p^3 + pq^2 + q^2}$$

since

$$pq^3 + pq < p^3q + pq \Rightarrow \left(\frac{A_5}{B_5}\right)^8 > \left(\frac{A_5}{B_5}\right)^6 .$$

From

$$\left(\frac{A_5}{B_5}\right)^7 = \frac{q^3 + p^2q + p}{p^3 + pq^2 + q}$$

rewritten as

$$\left(\frac{A_5}{B_5}\right)^7 = \frac{q^3 + 2p^2q + p^3 + pq}{p^3 + 2pq^2 + q^3 + pq}$$

$$\therefore \frac{q^3 + 2p^2q + p^3}{p^3 + 2pq^2 + q^3} > \frac{(q^3 + 2p^2q + p^3) + pq}{(p^3 + 2pq^2 + q^3) + pq}$$

$$\therefore \qquad \left(\frac{A_5}{B_5}\right)^8 > \left(\frac{A_5}{B_5}\right)^7 .$$

$f_5^9 = \langle 1,1,3,1 \rangle$; for the case $n = 4$, the form $f_4 = \langle 1,1,3 \rangle$ we have $A_1 = A_2 = 1$, $A_3 = 3$ and $A_4 = q^3 + p$.

$$\therefore \quad A_5 = q^4 + p(A_1q^3 + A_2q^2 + A_3q + A_4)$$

$$= q^4 + pq^3 + pq^2 + pq + pq^3 + p^2.$$

After simplifying we get

$$A_5 = pq^3 + p^2 + q$$

$$\therefore \quad B_5 = p^3q + q^2 + p$$

$$= p^3q + p^2 + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^9 = \frac{pq^3 + p^2 + q}{p^3q + p^2 + q} < 1 \text{ since } pq^3 < p^3q.$$

$$f_5^{10} = \langle 1,1,3,2 \rangle;$$

$$A_5 = q^4 + p(A_2q^2 + A_3q + A_4)$$

$$= q^4 + pq^2 + pq + pq^3 + p^2$$

$$= q^3 + pq^2 + pq + p^2$$

$$= q^2 + p$$

$$\therefore \quad B_5 = p^2 + q$$

$$= q^2 + p$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{10} = 1 .$$

$f_5^{11} = < 1,1,3,3 >$;

$$
\begin{aligned}
A_5 &= q^4 + p(A_3q + A_4) \\
&= q^4 + pq + pq^3 + p^2 \\
&= q^3 + pq + p^2 \\
&= q^3 + p
\end{aligned}
$$

$$\therefore \quad B_5 = p^3 + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{11} = \frac{q^3 + p}{p^3 + q} .$$

$f_5^{12} = < 1,1,3,4 >$;

$$
\begin{aligned}
A_5 &= q^4 + pA_4 \\
&= q^4 + pq^3 + p^2 \\
&= q^3 + p^2
\end{aligned}
$$

$$\therefore \quad B_5 = p^3 + q^2$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{12} = \frac{q^3 + p^2}{p^3 + q^2} .$$

This group from 9, 10, 11 and 12 the ratio $(A_5/B_5)^{12}$ is maximum. Clearly

$$\left(\frac{A_5}{B_5}\right)^{10} = 1 > \left(\frac{A_5}{B_5}\right)^{9} .$$

Since

$$p > p^2$$

$$\therefore \quad 1 > p^2 + q$$

$$p + q > p^2 + pq + q^2$$

$$(p + q)(p - q) > (p - q)(p^2 + pq + q^2)$$

$$p^2 - q^2 > p^3 - q^3$$

$$q^3 + p^2 > p^3 + q^2$$

$$\frac{q^3 + p^2}{p^3 + q^2} > 1 \Longrightarrow \left(\frac{A_5}{B_5}\right)^{12} > \left(\frac{A_5}{B_5}\right)^{10}$$

and

$$\left(\frac{A_5}{B_5}\right)^{11} = \frac{q^3 + p}{p^3 + q} = \frac{q^3 + p^2 + pq}{p^3 + q^2 + pq} < \frac{q^3 + p^2}{p^3 + q^2}$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{12} > \left(\frac{A_5}{B_5}\right)^{11} .$$

$f_5^{13} = < 1,2,1,1 >$; from the case $n = 4$, the form of $f_4 = < 1,2,1 >$ we have $A_1 = A_2 = 1$, $A_3 = q^2 + p$ and $A_4 = q^2 + pq^2 + p = p^2 + q + pq^2$

$$\therefore \quad A_5 = q^4 + p(A_1q^3 + A_2q^2 + A_3q + A_4)$$

$$= q^4 + pq^3 + pq^2 + pq^3 + p^2q + p^3 + p^2q^2 + pq.$$

After simplifying we have

$$A_5 = pq^2 + q^2 + p$$

$$\therefore \quad B_5 = p^2q + p^2 + q$$

$$= p^2q + q^2 + p$$

$$\left(\frac{A_5}{B_5}\right)^{13} = \frac{pq^2 + q^2 + p}{p^2q + q^2 + p} < 1 \text{ since } pq^2 < p^2q.$$

$f_5^{14} = < 1,2,1,2 >;$

$$A_5 = q^4 + p(A_2q^2 + A_3q + A_4)$$
$$= q^4 + pq^2 + pq^3 + p^2q + p^3 + p^2q^2 + pq$$
$$= q^3 + pq^2 + p^2 + pq + p^2q^2$$
$$= q^2 + p + p^2q^2$$
$$\therefore\ B_5 = p^2 + q + p^2q^2$$
$$= q^2 + p + p^2q^2$$
$$\left(\frac{A_5}{B_5}\right)^{14} = 1.$$

$f_5^{15} = < 1,2,1,3 >;$

$$A_5 = q^4 + p(A_3q + A_4)$$
$$= q^4 + pq^3 + p^2q + p^3 + p^2q^2 + pq$$
$$= q^3 + p^2 + p^2q^2 + pq$$
$$= q^3 + p + p^2q^2$$
$$\therefore\ B_5 = p^3 + q + p^2q^2$$
$$\therefore\ \left(\frac{A_5}{B_5}\right)^{15} = \frac{q^3 + p + p^2q^2}{p^3 + q + p^2q^2}\ .$$

$f_5^{16} = < 1,2,1,4 >;$

$$A_5 = q^4 + p(A_4)$$
$$= q^4 + p^3 + p^2q^2 + pq$$
$$= q^4 + p(p^2 + q) + p^2q^2$$
$$= q^4 + pq^2 + p^2q^2 + p^2 \text{ since } p^2 + q = q^2 + p$$

$$\therefore \quad B_5 = p^4 + p^2q + p^2q^2 + q^2$$

$$\left(\frac{A_5}{B_5}\right)^{16} = \frac{q^4 + pq^2 + p^2q^2 + p^2}{p^4 + p^2q + p^2q^2 + q^2} .$$

This group from 13, 14, 15 and 16, the ratio $(A_5/B_5)^{16}$ is the maximum. Clearly $(A_5/B_5)^{14} > (A_5/B_5)^{13}$.

Consider

$$1 > 1 - pq$$

$$1 > (p + q)^2 - pq = p^2 + pq + q^2$$

$$p + q > p^2 + pq + q^2$$

$$p^2 - q^2 > (p - q)(p^2 + q^2) + pq(p - q)$$

$$p^2 - q^2 > (p - q)(p + q)(p^2 + q^2) + pq(p - q)$$

$$p^2 - q^2 > p^4 - q^4 + p^2q - pq^2$$

$$q^4 + pq^2 + p^2 > p^4 + p^2q + q^2$$

and

$$q^4+pq^2+p^2+p^2q^2 > p^4 + p^2q + q^2 + p^2q^2$$

$$\therefore \frac{q^4 + pq^2 + p^2 + p^2q^2}{p^4 + p^2q + q^2 + p^2q^2} > 1 \Rightarrow \left(\frac{A_5}{B_5}\right)^{16} > \left(\frac{A_5}{B_5}\right)^{14}$$

$$\left(\frac{A_5}{B_5}\right)^{15} = \frac{q^3 + p + p^2q^2}{p^3 + q + p^2q^2}$$

$$= \frac{q^4 + pq^3 + p^2 + pq + p^2q^2}{p^4 + p^3q + q^2 + pq + p^2q^2}$$

$$= \frac{q^4 + pq^3 + p^2 + pq^2 + p^2q + p^2q^2}{p^4 + p^3q + q^2 + p^2q + pq^2 + p^2q^2}$$

$$= \frac{q^4 + pq^2 + p^2 + p^2q^2 + pq(q^2 + p)}{p^4 + p^2q + q^2 + p^2q^2 + pq(p^2 + q)}$$

$$< \frac{q^4 + pq^2 + p^2 + p^2q^2}{p^4 + p^2q + q^2 + p^2q^2}$$

since $pq(q^2 + p) = pq(p^2 + q)$. Thus

$$\left(\frac{A_5}{B_5}\right)^{16} > \left(\frac{A_5}{B_5}\right)^{15} .$$

$f_5^{17} = < 1,2,2,1 >$; from the case $n = 4$ the form $f_4 =$ $< 1,2,2 >$ tells us that $A_1 = A_2 = 1$, $A_3 = q^2 + p$ and $A_4 =$ $q^2 + p$.

$$\begin{aligned} \therefore \quad A_5 &= q^4 + p(A_1q^3 + A_2q^2 + A_3q + A_4) \\ &= q^4 + pq^3 + pq^2 + pq^3 + p^2q + pq^2 + p^2 \\ &= q^2 + pq^3 + p^2q + pq^2 + p^2 \\ &= pq^3 + p^2 + q^2 + pq \\ &= pq^3 + p^2 + q \end{aligned}$$

$$\begin{aligned} \therefore \quad B_5 &= p^3q + q^2 + p \\ &= p^3q + p^2 + q \end{aligned}$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{17} = \frac{pq^3 + p^2 + q}{p^3q + q^2 + p} < 1 \text{ since } pq^3 < p^3q.$$

$f_5^{18} = < 1,2,2,2 >$;

$$\begin{aligned} A_5 &= q^4 + p(A_2q^2 + A_3q + A_4) \\ &= q^4 + pq^2 + pq^3 + p^2q + pq^2 + p^2 \\ &= q^3 + pq^2 + pq(p + q) + p^2 \\ &= q^2 + pq + p^2 \end{aligned}$$

$$= p^2 + q$$

$$\therefore B_5 = q^2 + p$$

$$= p^2 + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{18} = 1 .$$

$f_f^{19} = < 1,2,2,3 >;$

$$A_5 = q^4 + p(A_3q + A_4)$$

$$= q^4 + pq^3 + p^2q + pq^2 + p^2$$

$$= q^3 + p^2q + pq^2 + p^2$$

$$= q^2 + p^2q + p^2$$

$$= q(q + p^2) + p^2$$

$$= q(q^2 + p) + p^2$$

$$= q^3 + pq + p^2$$

$$= q^3 + p$$

$$\therefore B_5 = p^3 + q$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{19} = \frac{q^3 + p}{p^3 + q} .$$

$f_5^{20} = < 1,2,2,4 >;$

$$A_5 = q^4 + p(A_4)$$

$$= q^4 + pq^2 + p^2$$

$$\therefore B_5 = p^4 + p^2q + q^2$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{20} = \frac{q^4 + pq^2 + p^2}{p^4 + p^2q + q^2}$$

$$= \frac{q^4 + pq^2 + p^2q + p^3}{p^4 + p^2q + pq^2 + q^3}$$

$$= \frac{q^4 + pq + p^3}{p^4 + pq + q^3} .$$

In this group from 17, 18, 19 and 20, the ratio $(A_5/B_5)^{20}$ is the maximum. Clearly $(A_5/B_5)^{18} = 1 > (A_5/B_5)^{17}$.

Consider

$$\begin{aligned}
pq &> 0 \\
p^2 + pq + q^2 &> p^2 + q^2 \\
(p - q)(p^2 + pq + q^2) &> (p - q)(p^2 + q^2) \\
p^3 - q^3 &> (p - q)(p + q)(p^2 + q^2) \\
&= (p^2 - q^2)(p^2 + q^2) \\
&= p^4 - q^4 \\
\therefore \quad p^3 - q^3 + pq &> p^4 - q^4 + pq \\
\therefore \quad \frac{q^4 + pq + p^3}{p^4 + pq + q^3} &> 1
\end{aligned}$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{20} > \left(\frac{A_5}{B_5}\right)^{18} .$$

Consider

$$\begin{aligned}
\left(\frac{A_5}{B_5}\right)^{19} &= \frac{q^3 + p}{p^3 + q} \\
&= \frac{q^3 + p^2 + pq}{p^3 + q^2 + pq} \\
&= \frac{q^4 + pq^3 + p^2q + p^3 + pq}{p^4 + p^3q + pq^2 + q^3 + pq} \\
&= \frac{q^4 + pq + p^3 + pq(q^2 + p)}{p^4 + pq + q^3 + pq(p^2 + q)}
\end{aligned}$$

$$< \frac{q^4 + pq + p^3}{p^4 + pq + q^3}$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{20} > \left(\frac{A_5}{B_5}\right)^{19} .$$

$f_5^{21} = < 1,2,3,1 >$; the form $f_4 = < 1,2,3 >$ having $A_1 = A_2 = 1$; $A_3 = q^2 + p$ and $A_4 = q^2 + p^2$.

$$\begin{aligned} A_5 &= q^4 + p(A_1q^3 + A_2q^2 + A_3q + A_4) \\ &= q^4 + pq^3 + pq^2 + pq^3 + p^2q + pq^2 + p^3 \\ &= q^3 + pq + pq^3 + pq^2 + p^3 \end{aligned}$$

$$\therefore B_5 = p^3 + pq + p^3q + p^2q + q^3$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{21} = \frac{q^3 + pq + p^3 + pq^3 + pq^2}{p^3 + pq + q^3 + p^3q + p^2q} < 1$$

since $pq^3 + pq^2 < p^3q + p^2q$.

$f_5^{22} = < 1,2,3,2 >$;

$$\begin{aligned} A_5 &= q^4 + p(A_2q^2 + A_3q + A_4) \\ &= q^4 + pq^2 + pq^3 + p^2q + pq^2 + p^3 \\ &= q^3 + pq^2 + pq + p^3 \end{aligned}$$

$$\therefore B_5 = p^3 + p^2q + pq + q^3$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{22} = \frac{q^3 + pq + p^3 + pq^2}{p^3 + pq + q^3 + p^2q} < 1 \text{ since } pq^2 < p^2q.$$

$f_5^{23} = < 1,2,3,3 >$;

$$A_5 = q^4 + p(A_3q + A_4)$$

$$= q^4 + pq^3 + p^2q + pq^2 + p^3$$

$$= q^3 + pq + p^3$$

$$\therefore B_5 = p^3 + pq + q^3$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{23} = 1 .$$

$$f_5^{24} = < 1,2,3,4 >;$$

$$A_5 = q^4 + pq^2 + p^3$$

$$= q^2 (q^2 + p) + p^3$$

$$= q^2 (p^2 + q) + p^3$$

$$= q^3 + p^2q^2 + p^3$$

$$\therefore B_5 = p^3 + p^2q^2 + p^3$$

$$\therefore \left(\frac{A_5}{B_5}\right)^{24} = 1.$$

This group from 21 to 24 are dominated by the maximum ratio from the other groups.

We have left to identify the maximum ratio among $(A_5/B_5)^4$, $(A_5/B_5)^8$, $(A_5/B_5)^{12}$, $(A_5/B_5)^{16}$, and $(A_5/B_5)^{20}$.

First let us prove the algebraical fact that if $a > 0$, $b > 0$, $c > 0$, and $d > 0$ then for $a > b$ and $c > d$, we have

$$\frac{a + c}{b + c} < \frac{a + d}{b + d} \tag{1}$$

and

$$\frac{a}{b} < \frac{a + c}{b + d} \text{ if } \frac{a}{b} < \frac{c}{d} . \tag{2}$$

Suppose

$$\frac{a + c}{b + c} < \frac{a + d}{b + d} \text{ where } c > d$$

$$\therefore (a + c)(b + d) \geq (a + d)(b + c)$$

$$ab + ad + bc + cd \geq ab + ac + bd + cd$$

$$d(a - b) \geq c(a - b)$$

and

$$(a - b) > 0$$

$$\therefore d \geq c \quad \text{contradiction.}$$

$\therefore$ if $a > b$ and $c > d$ then $a+c/b+c < a+d/b+d$ proved (1). From (2) suppose $a/b \geq (a+c)/(b+d)$ where $a/b < c/d$ $\therefore$ $ab + ad \geq ab + bc$. $a/b \geq c/d$ contradiction. $\therefore$ if $a/b < c/d$ then $a/b < (a+c)/(b+d)$ proved (2).

Consider the ratio

$$\left(\frac{A_5}{B_5}\right)^{16} = \frac{q^4 + pq^2 + p^2q^2 + p^2}{p^4 + p^2q + p^2q^2 + q^2}$$

$$= \frac{q^4 + pq^2 + p^2q^2 + p^2q + p^3}{p^4 + p^2q + p^2q^2 + pq^2 + q^3}$$

$$= \frac{q^4 + pq(p + q) + p^3 + p^2q^2}{p^4 + pq(p + q) + q^3 + p^2q^2}$$

$$= \frac{q^4 + pq + p^3 + p^2q^2}{p^4 + pq + q^3 + p^2q^2}$$

$$< \frac{q^4 + pq + p^3}{p^4 + pq + q^3} = \left(\frac{A_5}{B_5}\right)^{20}$$

which is greater than 1.

Consider the ratio

$$\left(\frac{A_5}{B_5}\right)^4 = \frac{q^4 + p}{p^4 + q}$$

$$= \frac{q^4 + pq + p^2}{p^4 + pq + q^2}$$

$$= \frac{q^4 + pq + p^3 + p^2q}{p^4 + pq + q^3 + pq^2}$$

$$> \frac{q^4 + pq + p^3}{p^4 + pq + q^3} = \left(\frac{A_5}{B_5}\right)^{20}.$$

Since by the result of (2) and

$$\frac{p^2q}{pq^2} = \frac{p}{q} > \frac{q^4 + pq + p^3}{p^4 + pq + q^3}$$

shown as follows: since

$$p^4 + p^3q + p^2q^2 + pq^3 + q^4 > 0$$

$$\therefore\ (p^4 + p^3q + p^2q^2 + pq^3 + q^4) + pq > pq$$

$$(p-q)(p^4+p^3q+p^2q^2+pq^3+q^4) + pq(p-q) > pq(p + q)(p - q)$$

$$p^5 - q^5 + p^2q - pq^2 > pq(p^2 - q^2)$$

$$p^5 + p^2q > p^3q-pq^3+q^5+pq^2$$

$$p^5 + p^2q + pq^3 > q^5 + pq^2 + p^3q$$

$$p(p^4 + pq + q^3) > q(q^4 + pq + p^3)$$

$$\frac{p}{q} > \frac{q^4 + pq + p^3}{p^4 + pq + q^3}$$

$$\therefore\ \left(\frac{A_5}{B_5}\right)^4 > \left(\frac{A_5}{B_5}\right)^{20} > \left(\frac{A_5}{B_5}\right)^{16}.$$

Consider the ratio

$$\left(\frac{A_5}{B_5}\right)^8 = \frac{q^3 + p^2q + p^2}{p^3 + pq^2 + q^2}$$

$$= \frac{q^3 + p^2 + p^2q}{p^3 + q^2 + pq^2}$$

$$> \frac{q^3 + p^2}{p^3 + q^2} .$$

Since by the fact of (2) and

$\frac{p^2q}{pq^2} = \frac{p}{q} > \frac{q^3 + p^2}{p^3 + q^2}$ shown as follows: we have

$$p^3 + q^3 > 0$$

$$p^2(1 - q) + q^2(1 - p) > 0$$

$$p^2 - p^2q + q^2 - pq^2 > 0$$

$$p^2 + q^2 > p^2q + pq^2 = pq(p + q) = pq$$

$$(p^2 + q^2)(p + q) > pq$$

$$(p^2 + q^2)(p + q)(p - q) > pq(p - q) \qquad ; p > q$$

$$p^4 - q^4 > p^2q - pq^2$$

$$p^4 + pq^2 > q^4 + p^2q$$

$$p(p^3 + q^2) > q(q^3 + p^2)$$

$$\frac{p}{q} > \frac{q^3 + p^2}{p^3 + q^2}$$

$$\therefore \left(\frac{A_5}{B_5}\right)^8 > \left(\frac{A_5}{B_5}\right)^{12} .$$

Rewritten

$$\left(\frac{A_5}{B_5}\right)^8 = \frac{q^4 + pq^3 + p^2q + p^2}{p^4 + p^3q + pq^2 + q^2}$$

$$= \frac{q^4 + pq(q^2 + p) + p^2}{p^4 + pq(p^2 + q) + q^2}$$

$$= \frac{q^4 + p^2 + pq(q^2 + p)}{p^4 + q^2 + pq(q^2 + p)} .$$

Consider that

$$1 > p$$

$$p > p^2$$

$$1 + p > 1 + p^2$$

$$1 > 1 - p + p^2$$

$$1 > 1 - 2p + p^2 + p$$

$$1 > q^2 + p$$

$$pq > pq(q^2 + p).$$

By using the fact from (1) and the result shown above we have

$$\frac{q^4 + p^2 + pq(q^2 + p)}{p^4 + q^2 + pq(q^2 + p)} > \frac{q^4 + p^2 + pq}{p^4 + q^2 + pq} = \frac{q^4 + p}{p^4 + q} = \left(\frac{A_5}{B_5}\right)^4$$

$$\therefore \left(\frac{A_5}{B_5}\right)^8 > \left(\frac{A_5}{B_5}\right)^4 .$$

Thus $(A_5/B_5)^8$ is the maximum ratio of the case $n = 5$ and this means $f_5^8 = < 1,1,2,4 > = f_5^*$ is the optimal algorithm and we have completed the proof.

APPENDIX B

```
//BOCC2894 JOB (1205,0530,ROX2),'CHODCHOEY SMC 2899'
// EXEC FORTCLG,REGION.GO=100K
//FORT.SYSIN DD *
C
C
C
C     THIS PROGRAM CONSISTS OF TWO PARTS, ONE IS
C     THE MAIN PART WHICH DETERMINING THE POSS
C    -IBLE ALGORITHM FOR A SPECIFIC VALUE OF N
C     FOR EACH OF THIS ALGORITHM, THE VALUE OF
C     A(N)/B(N)/ IS COMPUTED BY PART TWO OF THE
C     PROGRAM WHICH IS THE SUBROUTINE OF THIS
C     PROGRAM. FOR THE VALUE OF N=5,6,7 AND 8
C     THE PROGRAM WORKS OUT SATISFATORRILY  -
C     WITH TIME. BUT FOR N=9 AND 10 THE PROGRAM
C     IS TIME CONSUMING.
C               IN ORDER TO SOLVE OUT THIS
C    -PROBLEM , WE NEED TO ELIMINATE SOME   -
C     ALGORITHMS WHICH SEEMED NOT TO FIT THE
C     PATTERNS OF THE OPTIMAL ALGORITHM FOR
C     THE CASE OF N SMALLER THAN 10, SINCE ALL
C     THE SOLUTIONS SEEM TO HAVE DEFINITE PATT
C    -ERN OF OPTIMALITY ON A GIVEN VALUE OF P.
C     NOTE THAT THE PROGRAM DOWN BELOW CAN BE
C     USED FOR THE CASE OF N=10 ONLY, IF THE
C     CASE OF SMALLER N WILL BE RUN THEN SOME
C     CHANGES MUST BE MADE AS FOLLOW;
C
C       1. ELIMINATE STATEMENTS DO 10 IM=1,M-1
C
C       AND  R(M)=IM FROM THE MAIN PROGRAM FOR
C
C       ALL M=N+1,N+2,...,10
C
C       2. CHANGE STATEMENT  TRIM=A(10)/B(10)
C
C       TO   TRIM=A(N)/B(N)
C
C       3. AT SUBROUTINE PART, CHANGE STATEMENT
C
C       DO 10 I=3,10 TO  DO 10 I=3,N
C
C
C
      REAL*8 A(12),B(12),SMAX,TRIM,P,Y(12)
      INTEGER*4 R(12),G(12)
      READ (5,1000)Y
 1000 FORMAT (12F4.3)
      DO 100 K3=1,12
      P=Y(K3)
      DO 6 M=1,12
      R(M)=0
    6 G(M)=0
      SMAX=0.0D0
      R(2)=1
      DO 10 I3=1,2
      R(3)=I3
      DO 10 I4=1,3
      R(4)=I4
      DO 10 I5=1,4
      R(5)=I5
      DO 10 I6=1,5
      R(6)=I6
      DO 10 I7=1,6
      R(7)=I7
      DO 10 I8=1,7
      R(8)=I8
      DO 10 I9=1,8
      R(9)=I9
      DO 10 I10=1,9
      R(10)=I10
      CALL CALCAN (A,B,R,P)
```

```
      TRIM=A(10)/B(10)
      IF(TRIM.LE.SMAX) GO TO 10
      SMAX=TRIM
      DO 8 J=2,12
    8 G(J)=R(J)
   10 CONTINUE
      WRITE(6,1001)P
      WRITE (6,3000) SMAX
      WRITE (6,3001)(G(J),J=2,12)
 1001 FORMAT (' ',T10,'PROBABILITY P =',F6.3)
 3000 FORMAT (' ',T10,' A(N)/B(N) = ',F16.12)
 3001 FORMAT (' ',T10,' ALGORITHM ='//(' ',11I6),'0')
  100 CONTINUE
      STOP
      END
C
C     SUBROUTINE
C
      SUBROUTINE CALCAN(A,B,R,P)
      REAL*8 A(12),B(12)
      INTEGER*4 R(12)
      Q=1.0D0-P
      A(1)=1.0D0
      B(1)=1.0D0
      A(2)=1.0D0
      B(2)=1.0D0
      DO 10 I=3,10
      X1=0.0D0
      X2=0.0D0
      L1=R(I)
      L2=I-1
      DO 20 L=L1,L2
      X1=X1+P*A(L)*(Q**(L2-L))
   20 X2=X2+Q*B(L)*(P**(L2-L))
      A(I)=Q**(I-1)+X1
      B(I)=P**(I-1)+X2
   10 CONTINUE
      RETURN
      END
//GO.SYSIN DD *
C
C     INPUT DATA
C
.51 .55 .60 .65 .70 .75 .80 .85 .90 .95 .99 .999
C
C
C     FOR CASE OF N=10, THE PROGRAM ABOVE WILL
C     GIVE THE OUTPUT FOR EACH P IN THE FORM
C        PROBABILITY P = ...
C        A(N)/B(N)     = ...
C        ALGORITHM     = ...
C          .
C          .
C          .
C     UNFORTUNATELY, THE PROGRAM TAKES 0.5 HRS.
C     FOR EACH INPUT P, THAT MEANS WE HAVE TO
C     USE 7.5 HRS ALL TOGETHER. THE PROGRAM
C     BELOW ILLUSTRATED THE WAY TO DECREASE THE
C     RUN TIME BY ELIMINATING SOME ALGORITHMS
C     WHICH SHOULD NOT BE EVALUATED BECAUSE OF
C     THE UNCORRELATED PATTERN TO THE OPTIMAL
C     ALGORITHM OBTAINED IN THE CASE OF N  -
C     SMALLER THAN 10. THE RESULT WAS OBTAINED
C     IN ABOUT 7 SECONDS OF CPU. TIME ON ONE
C     VALUE OF P AND WAS CHECKED OUT CORRECTLY
C     WITH  THE CASE OF FULL COMBINATIONS WAS
C     USED FOR THE SAME P.
C
C
      REAL*8 A(12),B(12),SMAX,TRIM,P
      INTEGER*4 R(12),G(12)
      READ (5,1000)Y
```

```
1000 FORMAT (12F4.3)
     DO 6 M=1,12
     R(M)=0
   6 G(M)=0
     P=0.51D0
     SMAX=0.0D0
     R(2)=1
     R(3)=1
     R(4)=1
     DO 10 I5=1,4
     R(5)=I5
     DO 10 I6=1,5
     R(6)=I6
     DO 10 I7=3,6
     R(7)=I7
     DO 10 I8=4,7
     R(8)=I8
     DO 10 I9=6,8
     R(9)=I9
     R(10)=9
     CALL CALCAN (A,B,R,P)
     TRIM=A(10)/B(10)
     IF(TRIM.LE.SMAX) GO TO 10
     SMAX=TRIM
     DO 8 J=2,12
   8 G(J)=R(J)
  10 CONTINUE
     WRITE(6,1001)P
     WRITE (6,3000) SMAX
     WRITE (6,3001)(G(J),J=2,12)
1001 FORMAT (' ',T10,'PROBABILITY P =',F6.3)
3000 FORMAT (' ',T10,' A(N)/B(N) = ',F16.12)
3001 FORMAT (' ',T10,' ALGORITHM ='//(' ',11I6),'0')
 100 CONTINUE
     STOP
     END
C
C    SUBROUTINE
C
     SUBROUTINE CALCAN(A,B,R,P)
     REAL*8 A(12),B(12)
     INTEGER*4 R(12)
     Q=1.0D0-P
     A(1)=1.0D0
     B(1)=1.0D0
     A(2)=1.0D0
     B(2)=1.0D0
     A(3)=1.0D0
     B(3)=1.0D0
     A(4)=1.0D0
     B(4)=1.0D0
     DO 10 I=5,10
     X1=0.0D0
     X2=0.0D0
     L1=R(I)
     L2=I-1
     DO 20 L=L1,L2
     X1=X1+P*A(L)*(Q**(L2-L))
  20 X2=X2+Q*B(L)*(P**(L2-L))
     A(I)=Q**(I-1)+X1
     B(I)=P**(I-1)+X2
  10 CONTINUE
     RETURN
     END
C
C
C   RESULT OF THIS RUN IS ...
C       PROBABILITY P = 0.51
C       A(N)/B(N) = 1.1296
C       ALGORITHM = 1  1  1  3  4  5  7  8  9
C
C
```

APPENDIX C

```
//BCCC2894 JOB (1205,0530,ROX2),'CHODCHOEY SMC 2899'
C
C
C
C     THIS PROGRAM IS USED IN SEARCHING FOR
C     THE OPTIMAL ALGORITHM AT THE VALUE OF P
C     EQUALS .50+& FOR N=6,7,8,9 AND 10.
C           THE PROGRAM WAS PRESENTED IN THE
C     FORM WHICH CAN BE USED ONLY FOR THE CASE
C     OF N=10, FOR THE CASE OF SMALLER N WE
C     CAN PREPARE THE PROGRAM AS STATED IN  -
C     APPENDIX B
C
C
C
      REAL*8 A(10),B(10),SMAX,TRIM,P
      INTEGER*4 R(10),G(10)
      P=0.50D0
      DO 6 M=1,10
      R(M)=0
    6 G(M)=0
      SMAX=0.0D0
      R(2)=1
      DO 10 I3=1,2
      R(3)=I3
      DO 10 I4=1,3
      R(4)=I4
      DO 10 I5=1,4
      R(5)=I5
      DO 10 I6=1,5
      R(6)=I6
      DO 10 I7=1,6
      R(7)=I7
      DO 10 I8=1,7
      R(8)=I8
      DO 10 I9=1,8
      R(9)=I9
      DO 10 I10=1,9
      R(10)=I10
      CALL CALCAN (A,B,R,P)
      TRIM=B(10)/A(10)
      IF(TRIM.GE.SMAX) GO TO 10
      SMAX=TRIM
      DO 8 J=2,12
    8 G(J)=R(J)
   10 CONTINUE
      WRITE(6,1001)P
      WRITE (6,3000) SMAX
      WRITE (6,3001)(G(J),J=2,10)
 1001 FORMAT (' ',T10,'PROBABILITY P =',F6.3)
 3000 FORMAT (' ',T10,' B(N)/A(N) = ',F16.12)
 3001 FORMAT (' ',T10,' ALGORITHM ='//(' ',11I6),'0')
      STOP
      END
C
C
C     SUBROUTINE
C
C
      SUBROUTINE CALCAN(A,B,R,P)
      REAL*8 A(10),B(10)
      INTEGER*4 R(10)
      Q=1.0D0-P
      A(1)=1.0D0
      B(1)=0.0D0
      A(2)=1.0D0
      B(2)=0.0D0
      A(3)=1.0D0
      B(3)=0.0D0
    7 DO 10 I=3,10
      X1=0.0D0
      X2=0.0D0
```

```
      X(3)=0.0D0
      L1=R(I)
      L2=I-1
      DO 20 L=L1,L2
      X1=X1+P*A(L)*(Q**(L2-L))
      X2=X2+((L2-L)-1)*A(L)*(P**(L2-L))
   20 X3=X3+B(L)*(P**((L2-L)+1))
      A(I)=Q**(I-1)+X1
      B(I)=(I-1)*P**(I-2)+X2+X3
   10 CONTINUE
      RETURN
      END
```

LIST OF REFERENCES

1. Robbins, Herbert, "A Sequential Decision Problem With a Finite Memory," Proc. Nat. Acad. Sci., U.S.A., Vol. 42, p. 920-923.

2. Cover, Thomas, M., "Hypothesis Testing with Finite Statistics," The Annals of Mathematical Statistics, Vol. 40, No. 3.

3. Hellman, Martin E. and Cover, Thomas M., "Learning with Finite Memory," The Annals of Mathematical Statistics, Vol. 41, No. 3, p. 765-782.

4. Hellman, Martin E., "The Effects of Randomization on Finite Memory Decision Schemes," IEEE Transactions on Information Theory, IT-18, No. 4, p. 499-502.

5. Anderson, Calvin M., "Testing a Simple Symmetric Hypothesis by a Finite-Memory Deterministic Algorithm," MS Thesis, Naval Postgraduate School, Monterey, California, 1972.

6. Shubert, Bruno, "Evaluation of Certain Possibilities Associated with a Class of Markov Chains," Naval Postgraduate School, Report No. NPS55SY3111A.

7. Yakowitz, Sidney, "Multiple Hypotheses Testing by Finite Memory Algorithms," The Annals of Statistics, Vol. 2, No. 2, p. 323-336.

8. Shubert, Bruno, "Finite-Memory Classification of Bernoulli Sequence Using Reference Samples," IEEE Transactions on Information Theory, IT-20, No. 3, p. 384-387.

9. Sagalowicz, Daniel, "Hypotheses Testing with Finite Memory," Stanford University, Doctoral Degree Thesis, Palo Alto, California.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor Bruno Shubert (thesis advisor) Operations Research and Administrative Sciences Department Naval Postgraduate School Monterey, California 93940	1
4.	Professor Peter Wong (second reader) Mathematics Department Naval Postgraduate School Monterey, California 93940	1
5.	Chief of Personnel Operations ATTN: OPXC Department of the Army Washington, D. C. 20315	1
6.	2LT Boorapa Chodchoey Military Research and Development Center Klauy Num Tai, Sukumvit Rd. Bangkok, Thailand	2
7.	Department Chairman, Code 55 Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
8.	Chief of Naval Personnel (Pers-11b) Department of the Navy Washington, D. C. 20370	1

153761
Thesis
C4486 Chodchoey
c.1
Optimal deterministic algorithm for the simple symmetric hypotheses testing problem.